

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest





MAGYAR TUDOMÁNYOS AKADÉMIA
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE

EGYSÉGES KISSZÁMITÓGÉPES GÉPGYÁRTÁSTECHNOLÓGIAI
TERVEZŐRENDSZER VÁZLATOS RENDSZERTERVE KÜLÖNÖS
TEKINTETTEL A MONITOR RENDSZERRE

Irta:

Kovács Mihály

A kiadásért felel:

DR VÁMOS TIBOR

ISBN 963 311 031 9

TARTALOMJEGYZÉK

	Oldal
BEVEZETÉS	5
1. KISSZÁMITÓGÉPES RENDSZEREK GÉPIPARI ALKALMAZÁSÁNAK PROBLÉMAKÖRE	7
1.1 A feladat időszerűsége	7
1.2 Kiszámítógépek alkalmazásának szükségessége	8
1.3 Moduláris kiszámítógépes programozási rendszer követelményei	9
1.4 Kiszámítógéppel megoldandó feladatok	13
2. EGYSÉGES KISSZÁMITÓGÉPES TERVEZŐRENDSZER ÁLTALÁNOS FELÉPÍTÉSE	16
2.1 A rendszer általános kialakítása	16
2.2 A rendszer általános jellemzői	18
2.3 Az alrendszerek általános jellemzői	19
2.4 A teljes rendszer felépítése	22
2.5 Az alrendszerek felépítése	25
2.6 Bemenő nyelvek	33
2.7 Csatlakozási felületek	38
2.8 Adatbázis	39
2.9 A programozó nyelv	41
2.10 A monitor felépítése	42
3. A MONITOR RENDSZERVÁZLATA ÉS MŰKÖDÉSE	48
3.1 A monitor vezérlési rendszere	50
3.2 A monitor rendszer fő részei	57
3.3 A monitor rendszer használata	58
4. KÖVETKEZTETÉSEK	63
IRODALOMJEGYZÉK	64
"A" FÜGGELÉK	66

ÁBRAJEGYZÉK

Oldal

1. ábra	A rendszer általános kialakítása	17
2. ábra	Az egységes gépipari tervező rendszer elvi felépítése	23
3. ábra	Alrendszer vázlatos felépítése	26
4. ábra	"A" alrendszer létrehozásának és működtetésének elvi vázlata	30
5. ábra	A monitor elvi működése	43
6. ábra	A monitor elvi vázlata	45
7. ábra	A monitor vezérlési rendszere	49
8. ábra	A monitor vezérlés működése	52
9. ábra	A monitor vezérlési szintjei	54
10. ábra	A monitor rendszer elhelyezkedésének vázlata a kisszámítógépen	59
11. ábra	R10 tervezői konfiguráció	68

BEVEZETÉS

Az egységes kisszámitógépes gépgyártástechnológiai tervezőrendszer alapelvei és rendszertechnikai felépítése hasonló a nagyszámítógépekre elképzelt rendszerekéhez. Így alapvető jellegzetességeit meghatározza az a tanulmány, amely a gépipari AMT tudományos főirányait és alapelveit rögzíti [1].

Jelen tanulmányban tovább haladunk a [1]-ban jelölt uton, nem csak abban, hogy kisszámitógépes jellegzetességekre térünk ki, hanem abban is, hogy igyekszünk vázlatos rendszertervet készíteni az egész rendszer keretével szolgáló monitor rendszerre. A monitor rendszert nem lehet a teljes rendszer nélkül tárgyalni. Ezért szükséges a teljes kisszámitógépes tervező rendszerre vonatkozó elképzeléseket is valamivel határozottabb formába önteni.

Az első fejezet a tervezőrendszerek kisszámitógépes alkalmazásának problémakörébe vezet be. A számítógép konfigurációval kapcsolatos megfontolásokat az "A" függelék tartalmazza.

A második fejezet az egységes kisszámitógépes tervezőrendszer általános felépítésével foglalkozik.

A harmadik fejezet a monitor rendszer vázlatos kialakítását tárgyalja és bemutatja a monitor rendszer működését.

Végül röviden összefoglaljuk egy részletes rendszerterv feladatait.

Az előzetes rendszerterv összeállításának kialakításában a SzTAKI számos specialistája volt segítségemre.

Köszönetet kívánok mondani Bródy Ferenc, Gerhardt Géza, Farkas Ernő, Gyürki József, János József, Kardos Péter, Kovács Erika, Krammer Gergely, Párkányi Tamás és Szokoly Pál

kollegáim tanácsaiért.

Továbbá külön köszönet illeti a GTI részéről Szilágyi István főosztályvezető és Cser István tud. csoportvezető elvtársakat, akik az ügy fontosságát felismerve e téma hivatalos indítását kezdeményezték.

Ugyancsak köszönetet kívánok mondani dr. Hatvany József tudományos osztályvezetőnek és dr. Horváth Mátyás tudományos tanácsadónak, akik e munkát lehetővé tették és irányították.

1. / KISSZÁMITÓGÉPES RENDSZEREK GÉPIPARI ALKALMAZÁSÁNAK PROBLÉMAKÖRE

E problémakör igen sokoldalú. Sem itt, és a további fejezetekben sem vállalkozhatunk a teljesség igényével való tárgyalásra. Néhány jelentősebb tényezőre viszont szeretnénk felhívni a figyelmet. Megjegyezzük, hogy az eszköz háttérrel az "A" függelék foglalkozik.

1.1 A feladat időszerűsége

A fejlett ipari országok gyakorlatából közismert, hogy a kisszámitógépek polgárjogot nyertek a gépgyártó üzemek napi termelési feladatainak ellátásában. Ez a tény nemcsak a műszaki adminisztrációra, a tervezés egyes fázisaira, a technológiai előkészítés különböző lépcsőire, a műhelyszintű termelés-irányításra vonatkozik, hanem még a közvetlen gépirányításra is.

Az integrált gyártáselőkészítési és gépgyártási rendszerek hazai bevezetése napirendre tűzte azt az igényt, hogy a kisszámitógépek műszaki, gépipari alkalmazása terén folyó kutatási- fejlesztési munkákat is összefogjuk.

Lehetőség szerint olyan egységes rendszerszemléletet alakítsunk ki, amely közös keretbe fogja össze az eddig többé-kevésbé elszigetelten folyó tevékenységeket, egyszersmint teremtsen meg az alapot a további egységes fejlesztéshez.

Több alapnak tekintett [2,3] tanulmány felmérte az új integrált gyártáselőkészítő és az integrált gépgyártó rendszerek programellátásával kapcsolatos igényeket. Feltárta azokat a körülményeket, amelyek egy hosszútávú fejlesztő munka alapjául szolgálnak. Ugyanakkor kimondja, hogy a feladatok nagyszámú megoldása mellett gondoskodni kell a kisszámitógépes megoldásról is. A kisszámitógépes rendszer számítástechnikai

alapjául is minden szempontból az MSZR rendszert kell tekinteni.

Egységes kisszámitógépes programozási rendszer létrehozása a gépgyártás területén nem csak az integrált rendszerek miatt célszerű. Ilyen rendszer használható minden hagyományosnak tekintett feladat megoldásához is, mint monitor rendszer. Napjainkban, amikor a legkülönbözőbb területeken szinte gomba módra szaporodnak a kisszámitógépes alkalmazások és a szándékolt, műszaki jellegű program fejlesztésének, különösen idősebb ezeket egységes keretbe foglalni. Ezzel elháríthatók a felesleges párhuzamosságok, a költségek jelentősen csökkenthetők, a munkák meggyorsíthatók.

Ezt a feladatot napirendre tűzi a hazai kisszámitógép gyártó ipar és a kisszámitógépeket alkalmazó gépipari vállalatok közös igénye az, hogy egységes és bőséges program háttér biztosítsa mielőbb a kisszámitógépek ipari alkalmazását. A hazai igények mellett vannak a szocialista országok együttműködéséből ránk háruló feladataink is ezen a téren, amelyeket teljesíteniünk kell a magunk érdekében is.

1.2 Kisszámitógépek alkalmazásának szükségessége

Kisszámitógépek alkalmazására a gépipari termelésben nem csak a mai számítástechnikai és gazdasági viszonyok között van szükség, hanem a jövőbeli, fejlett, nagy rendszerekben is szükség lesz.

Ennek alapvető oka az ember-gép párbeszéd, a grafikus párbeszéd és a programmegszakítás lehetőségének igénye. A gépgyártási jellegű feladatok jelentős részénél olyan munkát kell végezni a számítógéppel, amely nehezen, vagy egyáltalán nem sematizálható. Az emberi kezelőnek, a konstruktőrnek, technológusnak vagy diszpécsernek kell a gép szolgáltatotta eredményeket értékelni és döntéseket hoznia. A lekérdezhető közbülső eredmé-

nyek elégtelen volta esetén meg kell szakitania a programfutásokat, lehetőségének kell lennie a tetszőleges számú ismétlésre. A feladatok természete legtöbbször olyan, hogy variánsokat kell képezni és a legkedvezőbbet kiválasztani. Erre a célra a kisszámitógép - esetleg grafikus perifériákkal kiegészítve - önállóan is, vagy nagyszámítógéphez csatolva szatellita üzemben is kiválóan alkalmas.

Ugyanakkor a számítástechnikai piac jelenlegi és várható helyzetét is célszerű figyelembe venni. Ebből kiderül, hogy a gépgyáraknak műszaki célokra elsősorban kisszámitógépek beszerzésére van lehetősége. Jobb esetekben ezeket csatolni tudják nagyobb számítógéphez. Ezért a [3] tanulmány a fejlesztés alapelveinek tekinti, hogy a létrehozandó programok számítástechnikai igénye minimális legyen és azok kisszámitógépen is futtathatók legyenek.

Valóban, ha azt akarjuk, hogy az új termeléselőkészítési, technológiai folyamatirányítási módszerek széles körben elterjedjenek, akkor ki kell alakítani a kisszámitógépes programrendszereket is. Ezzel, vagyis az alacsonyabb beruházási költség lehetővé tételével a gépgyárak többsége számára a kialakítandó rendszerek elérhetőbbek lesznek.

Ugyanakkor a továbbfejlesztés elől sem zárja el az utat a kisszámitógépek alkalmazása, mert ezekből később központi számítógép irányításával tetszőleges hierarchikus nagyrendszerek képezhetők.

1.3 Moduláris kisszámitógépes programozási rendszer követelményei

A feladat olyan projekt előkészítése és megtervezése, amely a gépgyártás területén felmerülő, kisszámitógéppel megoldható problémákat egységes programozási rendszer keretében oldja meg.

A szerteágazó és elszigetelt programozási munkák később csak mint tapasztalatok hasznosíthatók, de eredményeik csak igen csekély mértékben vihetők tovább. A feladatok természete olyan - különösen az integrált gyártórendszerek megjelenése óta - , hogy egy összefüggő egész felé mutatnak, teljességet csak később, a jövőbeli nagy rendszerekben nyernek.

Ezért nyilvánvaló, hogy a technológiai folyamatok előkészítése és irányítása kisszámitógépes munkáit is olyan megbízható alapra kell építeni, amely egyrészt biztosítja, hogy a mai eredményeket évek múlva is felhasználhassák, másrészt lehetővé teszi az együttes felhasználást, az integrálást bármely közbülső fokon is. Együttal megkönnyíti, megrövidíti az alkalmazói programcsomagok írását azzal, hogy közös programozási alapot, keretet, szervezést nyújt a különböző és esetleg érintkező feladatok programjainak.

A követelmények meghatározásakor nem hallgathatjuk el azokat a sokszor áthidalhatatlannak látszó nehézségeket sem, amelyekkel a kisszámitógépek gyorsan fejlődő és gyakran változó világában szembe kell nézni.

A számítógép típusok gyakori változása, a még kiforratlan konstrukciók megbízhatósági problémái, a gyenge alapsoftware ellátottság, a szervezetlenség és a gyakori bizonytalanság léghőre az, amellyel e hosszútávú fejlesztés megkezdésekor számolnunk kell.

Azonban, ha meg akarjuk oldani a gépgyártás területén feladatainkat, számolva a napi valósággal, olyan rendszert kell kialakítanunk, amely a korlátozó tényezőket minimálisra csökkenti. A változó számítógép típusokkal és programozásukkal szemben rendszerünk adaptációs képességét kell fokozni, minimálisra szorítva le a rendszer azon részeit, amelyeket különböző géptípusok és gépelrendezések esetén újra kell írni.

Az említett nehézségek nagyjából négy alapvető fontosságú témakör kidolgozásával látszanak elkerülhetőnek.

Meg kell határozni a gépgyártás területén, azokat a feladatcsoportokat, amelyeket kisszámitógépes rendszerekkel kívánunk megoldani.

Ezeket a feladatokat fel kell bontani, és teljes mélységben specifikálni kell minden résztvevő tevékenységet. A körültekintően elvégzett felbontás eredményeképpen kialakítható egy moduláris felépítésű rendszer, amelybe az egyes feladatokat illetve részfeladatokat megoldó programok építőközként rakhatók be. Az alkalmazási terület szükségleteinek megfelelően az építőközből tetszőleges rendszer alakítható ki. Célszerű a különálló feladatcsoportokat megoldó egységeket önálló alrendszerekként kezelni, amelyeket a részfeladatokat megoldó modulokból lehet kiépíteni. A modulok esetleges további tagolásának sem lehet akadálya. Az alrendszerek között és az alrendszerekben belül a modulok közötti csatlakozási felületeket szabványosítani kell. Így elérhető a különböző helyeken kidolgozott programok későbbi csatlakoztatása és elkerülhető az újbóli megírás vesztesége.

Létre kell hozni az egész moduláris rendszerre kiterjedő keretrendszert, amelybe az egyes alrendszerek moduljai beilleszthetők, és amely később biztosíthatja az alrendszerek közötti információcserét, az alrendszerek összehangolt munkáját. Ez a keretrendszer leginkább egy speciális célu, problémaorientált operációs rendszerhez hasonlítható.

A keretrendszer, amelyet monitor rendszernek is nevezünk, azon kívül, hogy elvégzi a hagyományos operációs rendszer funkciókat, mindazon szolgáltatásokat is nyújtja, amelyekre a problémamegoldó moduloknak szükségük lehet.

Tehát a problémamegoldó modulokat kell illesztenie az adott számítógép konfigurációhoz, ezzel egyszersmind egyszerűsíti a feladatmegoldó programok kidolgozási munkáit. A keret adó monitor rendszer maga is moduláris felépítésű, hogy egy alapvető magból tetszés szerinti építőelemekkel legyen bővíthető a mindenkori kívánalmaknak megfelelően.

Igen nehéz feladat, a nagy változatosságot mutató kisszámítógépes területen - ahol a gépek jórésze még nem alkalmas az általánosan elterjedt programozási nyelvek hatékony futtatására sem - olyan módszert találni, amellyel a kidolgozott programok áttelepíthetősége biztosítható. Meg kell találni, illetve ki kell fejleszteni azt a technikát, a problémamegoldó programok és a monitor rendszer olyan dokumentálását, amely a programok átvitelét egyik kisszámítógépről a másikra a legkevesebb erőfeszítés mellett biztosítja. Ezen a területen minden eredmény igen nagy jelentőségű, hiszen a gépgyártás technológiai folyamatai előkészítésére és irányítására szolgáló programrendszer kidolgozása olyan hosszutávú feladat, amelyen belül még számítógépes generációváltás is bekövetkezhet. Tehát meg kell közelíteni a számítógépfüggetlen programírás és a programdokumentálás lehetőségét úgy, hogy a programok bármikor könnyen alkalmazhatók legyenek a számba jöhető lényegesebb kisszámítógépekre.

A feladat nagysága által meghatározott hosszutávú programfejlesztésbe nem lehet a siker reményében anélkül belefogni, hogy ne legyenek lerögzítve azok a közös módszerek, szabályok, amelyek biztosítják a programkészítés szakszerűségét, egységességét.

Az egyöntetűség biztosítására megfelelő háziszabványokat kell előírni, amelyek minden programkészítőre kötelezőek. Ki kell alakítani a megbízható és termelékeny programírás módszereit. Nagy gondot kell fordítani a dokumentálhatóságra és a dokumentálásra is.

Megfelelő kritikával ki kell választani és meg kell honosítani azokat a fejlett programirási módszereket, amelyek biztosíthatják egy nagy teljesítményű programrendszer lépcsőről lépésre való kiépítését, megbízható üzemeltetését, a könnyű karbantartás és továbbfejlesztés lehetőségét.

1.4 Kisszámitógéppel megoldandó feladatok

A következőkben rövid felsorolást adunk a teljesség igénye nélkül azokról a témakörökről, amelyekben már most is világos, hogy szükség és igény van a kisszámitógépes megoldásokra.

Moduláris kisszámitógépes NC programozó rendszer

Szükséges a közepes számítógépre már kidolgozás alatt álló egységes modulrasi KGST NC programozási rendszer kisgépes megfelelőjének kialakítása. Ebbe a munkába már célszerű belevinni az interaktív programkészítés és a grafikus megjelenítés lehetőségét is.

Általános NC vezérlőszalag kezelő és ellenőrző rendszer

Olyan kisszámitógépes rendszert is célszerű kialakítani, amely alkalmas az NC vezérlőszalagok közvetlen vezérműködésben való megírására, a hagyományos kézi programozás számítógépes segítésére, a vezérlő szalagok kezelésére, javítására, konvertálására és ellenőrzésére. Ezen belül mindazon szolgáltatásokra képes, amelyekre egy gyári NC programirodában szükség lehet.

Nagygépes munkák kisszámitógépes kiegészítése

Nagygépes technológiai programozási rendszerek kisszámitógépes kiszolgálása. Előfeldolgozások megoldása kisszámitógépen. Különböző nagygépes munkák közbenső eredményeinek el-

lenőrzése, módosítása, utófeldolgozása kisszámitógéppel.

Technológiai_folyamatirányítási_programrendszer

Szükség van a meglévő szerszámgépirányító programok általánosabbá tételére és bővítésére a munkadarab és szerszámcse-re automatizálásának következtében. Bővíteni, fejleszteni kell azokat különösen a diagnosztikai, karbantartási képességek irányában. A szerszámkopások figyelése, a szerszámkarbantartási felügyelet már ma is reális igény.

Automatikus_termelésirányítás

Ki kell alakítani folytonos jelleggel egy integrált gyártórendszer minden elemére kiterjedő finomprogramozást. Biztosítani kell az irányítási döntések automatizálhatóságát is.

Optimalizáló_termelésprogramozás

Az igen nagy beruházott értéket képviselő integrált gyártórendszerek felvetik az extenzív kihasználást jobban biztosító, optimalizálható termelésprogramozási rendszer kifejlesztését, amely alkalmas technológiai variánsokból termelési programvariánsok képzésére és az optimális variáns kiválasztására.

Technológiai_folyamattervező_rendszer

Az állandóan változó alkatrészválaszték megmunkálásának megtervezéséhez meg kell oldani a gépválasztás, a megmunkálási mód és sorrend meghatározása, a szerszámválasztás, az optimális forgácsolási paraméterek meghatározásának feladatát. Ezt kézi munkával ma már lehetetlen ésszerű határokon belül elvégezni. E feladat egyes részei megoldhatók kisszámitógéppel.

Vállalati technológiai adatbank

Az átfogó technológiai adatbázis létesítése megkezdődött. Központi nagygépes adatbank kifejlesztése megindult. Ennek kihelyezett, vállalati, vagy műhelyszintű megfelelőjére olyan kisszámitógépes rendszert célszerű kialakítani, amely a központi adatbankkal rendszeres kapcsolatban van, és amely ki tudja szolgálni a helyi technológiai igényeket.

Forgácsoló szerszámtervező rendszer

Ez a rendszer arra alkalmas, hogy az adatbázisból, a meglevő készletből válasszon e feladatnak megfelelő szerszámot és szerszám befogó készüléket. Ha nincs megfelelő, akkor lehetőséget nyújt interaktív szerszámtervezésre és gyors dokumentálásra.

Palettás felfogó készülék tervezőrendszer

Lehetőséget nyújt, hogy a viszonylag kis elemszámból álló összerakható készülékelemekből az automatikus dokumentálással együtt, a munkadarabnak megfelelő befogó, leszorító készüléket tervezhessenek.

2. EGYSÉGES KISSZÁMITÓGÉPES TERVEZŐRENDSZER ÁLTALÁNOS FELÉPÍTÉSE

A gépipari tervezési feladatok, így a technológia tervezőfeladatok is meglehetősen összetettek. Az ilyen jellegű feladatok sokrétősége miatt lehetetlen olyan rendszert készíteni, amely egyszer és mindenkorra megoldja az összes problémát.

Amit lehet és ésszerű kialakítani az egy olyan rugalmas mérnöki rendszer, amelyet az építőköcka elv alapján egységes rendszeren belül lehet fokozatosan kiépíteni és tetszőlegesen bővíteni.

Munkánkban olyan megoldásokra alapoztunk, amelyek e fenti elvet követték. Így felhasználtuk az amerikai MIT-ben /Massachusetts Institute of Technology/ kidolgozott ICES /Integrated Civil Engineering System/ rendszer [4] , a szovjet IK AN ESzSzR-ben /Insztitut Kibernetiki Akademii Nauk-Esztonszkoj SzSzR/ kidolgozott ISzP /Integrirovannaja Szisztéma Programirovaniye/ rendszer [5] , és a KGST szerszám-gépipari egyezményének keretében készült "Egységes moduláris NC programozási rendszer" [6] eredményeit.

2.1 A rendszer általános kialakítása

A rendszer csoportokra bontott feladatokat képes megoldani az egyes feladatköröket megoldó alrendszerei segítségével.

Egy időben csak egy alrendszer működik. A rendszer működése egyenlő az alrendszerek működésének egymásutánjával.

Az alrendszereket - egy időben mindig csak egyet - a központi monitor rendszer irányítja az alkalmazó parancsainak megfelelően.

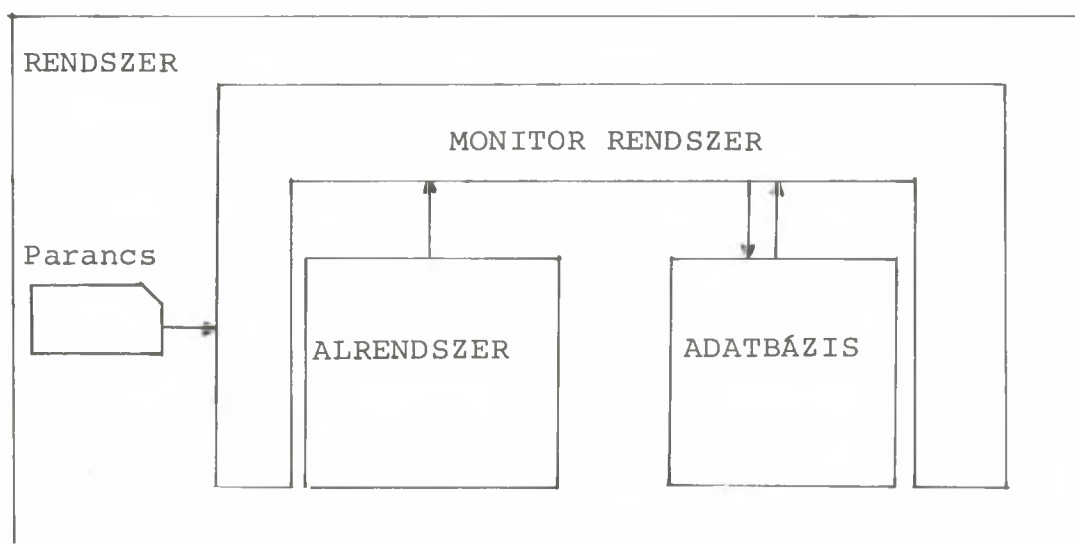
Az alkalmazó parancsait egységes szerkezetű mérnöki beme-

nő nyelveken adhatja be. Minden alrendszer rendelkezik saját - feladatköréhez illesztett - bemenő nyelvvel. Ezen kívül a rendszer munkájának irányítására létezik egy, az összes alrendszerre nézve közös rendszerkezelő nyelv is.

Az alrendszerek működésük során igénybe vehetik az egységes közös adatbázist. Ez az adatbázis rugalmas, egy alrendszerhez vagy alrendszerek egy csoportjához előre képezhető a feladatnak megfelelő adatbázis.

Az alrendszerek közötti együttműködést az egységes csatlakozási felületek biztosítják. Ezek szabványos fájl formátumok, amelyek lehetővé teszik, hogy az előző alrendszer eredményeit a soron következő felhasználhassa.

Az 1. ábra mutatja a rendszer általános elvi felépítését, a monitor rendszer, egy aktuális alrendszer és az adatbázis kapcsolatát.



1. ábra

A rendszer általános kialakítása

Megjegyezzük, hogy ez az ábra csak bevezető jellegű. Lényegében azt tükrözi, hogy a rendszerben minden tevékenységet a monitor rendszer koordinál. Az alkalmazó is csak a monitor rendszeren keresztül érheti el a rendszer különböző egységeit.

A rendszer felépítését úgy tárgyaljuk, hogy fokozatosan haladunk a részletek felé.

2.2 A rendszer általános jellemzői

A rendszer a műszaki alkalmazót két irányban segíti.

Az egyik a mérnök technológus, aki csak a műszaki feladat közvetlen megoldásában érdekelt. Az ő számára ez kifejezetten futtatási rendszer.

A másik alkalmazó az alrendszerépítő rendszerprogramozó. A rendszer az ő számára is igyekszik olyan eszközöket nyújtani, amelyek munkájának jó részét automatizálják.

A technológus számára biztosított kényelem az, hogy a rendszer kezeléséhez nem szükségesek részletes számítógépes ismeretek. Csak a rendszerkezelő nyelvvel kell megismerkednie és annak a témakörnek feladatleíró nyelvével, amely megfelel a speciális műszaki feladatkörének. E két nyelv sem teljesen eltérő, mert a rendszerben egységes szintaktikai szabályokat használunk.

A rendszerprogramozó számára elsőrendű az, hogy a rendszer minden szintjén érvényesül a modularitás elve. A rendszer tetszőlegesen bővíthető. Speciális feladatokra könnyen összeállíthatók az alrendszerek. Mindezekhez még külön segítséget is nyújtanak a rendszer speciális szolgáltatásai.

A rendszerben fontos szerepet kap a programozási munka egyszerűsítése és egységesítése.

A rendszer biztosítja, hogy a programozói munkák zömét kitévő modulokat a számítógépes környezettől függetlenül lehessen megírni. Ez azzal éri el, hogy a monitor rendszerbe helyezi mindazokat a programokat, amelyek számítógép és operációs rendszer függőek.

A kisszámítógépre való programozás egységesítése, hatékonyságának javítása, méginkább áttelepthetősége érdekében speciális kisszámítógépes programozó nyelv készül [7] .

A kisszámítógéppel való gazdaságos programfuttatás, valamint a nagyterjedelmű műszaki feladatok ellentmondásának feloldására a rendszer biztosítja az operatív memória jó kihasználását. Ehhez olyan eszközöket nyújt, amelyek a programozó munkáját is megkönnyítik.

Lehetővé teszi a dinamikus programkezelést és a dinamikus adatkezelést. Ebben a monitor rendszernek együtt kell működni a [7] -ben kialakítandó magasszintű programozó nyelvvel és felhasználja a [9] -ben kipróbált megoldásokat.

Lényeges jellemzője a rendszernek, hogy egy-egy feladat megoldását többféle módon is lehetővé teszi, mintegy megoldási választékot nyújt az alrendszerépítő rendszerprogramozó és a moduliró programozó számára, amelyekből kiválasztható a célnak leginkább megfelelő megoldási variáns.

2.3 Az alrendszerek általános jellemzői

A mérnöki tevékenység jól megkülönböztethető fázisokra bontható. A mérnök munkája ezeknek a szakaszoknak az egymásutánjából áll. Az egyes munkafázisokat lehet külön - külön is automatizálni, de később biztosítani kell az egymással kapcsolatban levő munkafázisok összehangolását. Ez csak úgy lehetséges, ha már eleve egységes szemlélet alapján dolgozzuk ki az egyes fázisok automatizálását, ezért az egységes rendszert

alrendszerekre bontjuk. Ezek az alrendszerek megfelelnek a mérnöki munka fázisainak. Az alrendszerek felépítése olyan, hogy alkalmasak az önálló működésre. Ezen kívül jellemzőjük, az is, hogy lépésről lépésre fejleszthetők, bővíthetők.

A rendszer felépítéséből következik, hogy az alrendszerek együttműködésére több lehetőség is rendelkezésre áll. A rendszerprogramozó, a csatlakozási felület tervező ezekből válogathat.

Az egyes alrenderek különböző logikai, rendszertехnikai és funkcionális egységekre tagolhatók. Ezek részletezésére később térünk ki.

Az egyes mérnöki feladatkörökön az alrendszereken belül megkülönböztethetők különálló feladatcsoportok, feladatok és részfeladatok is. A részfeladatokra bontás a moduláris felépítés műszaki alapja.

A különálló, de kapcsolódó részfeladatokat megoldó modulokból fokozatosan kiépíthetők és tetszés szerint bővíthetők az alrendszerek. Így lehetővé válik a változó körülményekhez való alkalmazkodás. Némi egyszerűsítéssel a helyi viszonyoknak megfelelő speciális változatok gyors készítése a meglevő modulkészletből való válogatásra, egyes speciális modulok megírására korlátozódik.

A teljes rendszer szervezését a monitor rendszer végzi. Ezen belül a monitor rendszer minden egyes alrendszernek közös vázszerkezete is. Az alrendszert is a monitor rendszer szervezi és irányítja. Mivel a mérnöki munka fázisai egymás után következnek, a monitor rendszer mindig az aktuális alrendszert szolgálja ki.

Az egységes rendszer megkönnyíti az egyes alrendszerek kifejlesztését. Egy sor olyan szolgáltatást nyújt, amelyről az alrendszer készítőinek már nem kell külön-külön gondoskodniuk, csak alkalmazniuk kell a rendszer speciális eszközeit.

Az alrendszerek munkájának összehangolására a monitor rendszer önmagában nem elegendő. A monitor rendszer csak annak lehetőségét teremti meg, hogy az alrendszerek bemenő adataikat a kívánt helyről vagy helyekről kapják. Ez lehet egy előzőleg működött alrendszer kimenete, vagy közbenső eredménye is. Ugyancsak lehetőséget ad arra, hogy egy alrendszer az eredményeit a kívánt helyre tegye le további felhasználásra. Mindezek közben biztosítja, hogy az alrendszer az adatbázissal is kommunikáljon. Az alrendszerek együttműködését csak a szigorúan definiált csatlakozási felületek biztosíthatják.

Külön kérdést jelent, hogy mit tekintünk alrendszernek. Ezt a fogalmat rugalmasan kell kezelni. A rendszer lehetővé teszi, hogy bármilyen szűk feladatcsoportot, szélső esetben egyetlen feladatot vagy egyetlen programot, alrendszerként kezeljen. Mint látni fogjuk, kisszámitógép esetén különösen fontos a megfelelő lebontás az, hogy egy alrendszer minél kisebb részekből álljon és ez nem csak a speciális műszaki feladatmegoldó modulokra, programokra áll, hanem az alrendszer egyéb részeire, szöveg és adatbázisaira is. Ez a futás gyorsítása szempontjából is jelentős lehet.

Ezért az alrendszer olyan, hogy állhat több, alacsonyabb szintű alrendszerből is. A monitor rendszer azonban nem tesz különbséget a különböző szintű alrendszerek között. Minden alrendszert egyformán kezel. Ezt a tulajdonságot és a többi lehetőséget az alrendszer készítőjének kell figyelembe venni a legkedvezőbb felépítés kialakítására.

2.4 A teljes rendszer felépítése

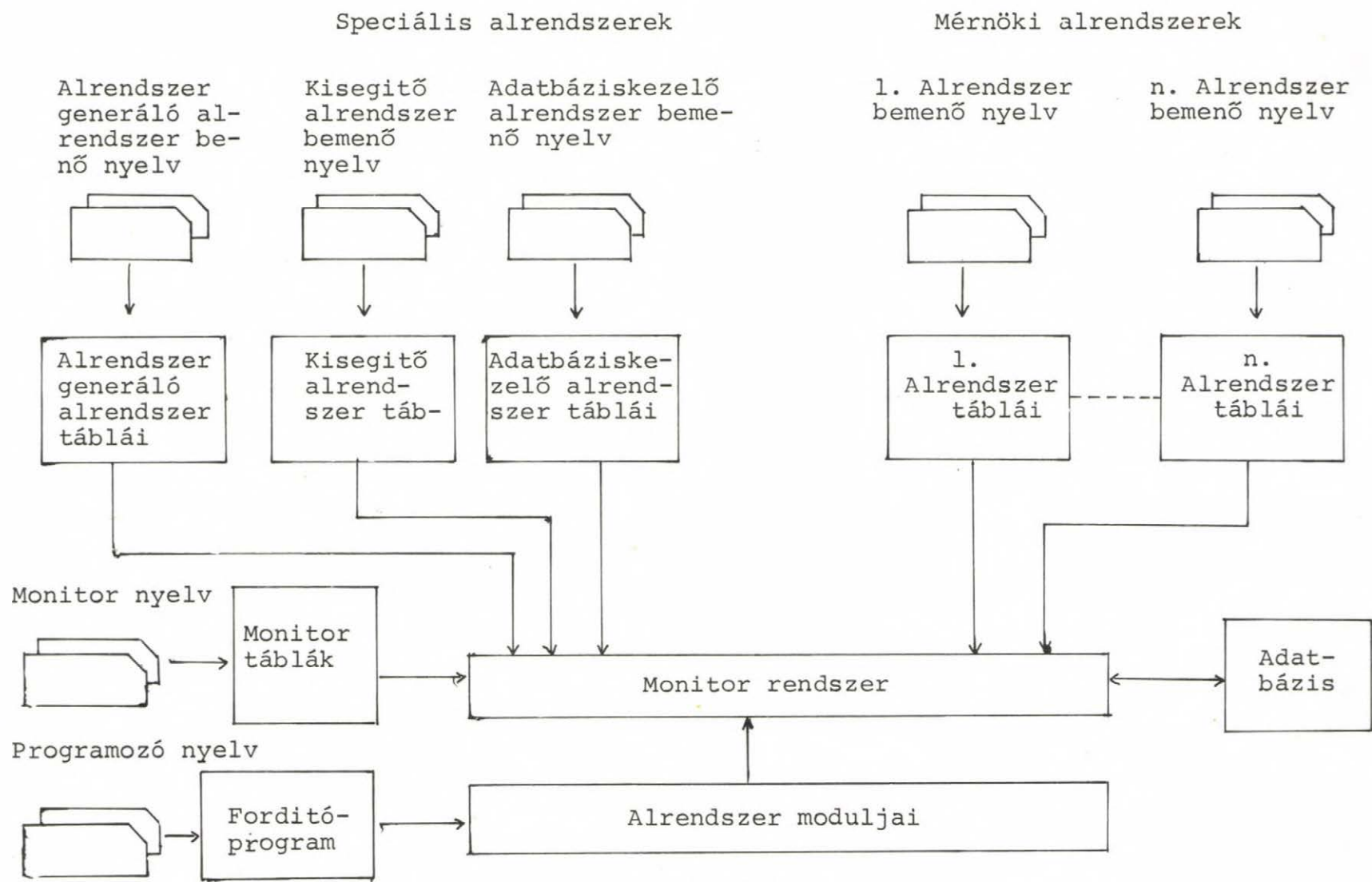
A teljes rendszer elvi vázlata a 2.sz. ábrán látható. Az ábra a rendszer alapvető elemeit tartalmazza.

A teljes rendszer magja a monitor rendszer. Ehhez kapcsolódnak a különböző alrendszerek, és ez tart kétirányu kapcsolatot a rendszer adatbázisával. Bár minden egyes alrendszer a monitor rendszerhez kapcsolódik, mint az 1. ábrán is látható, a monitor egyszerre mindig csak egy alrendszerrel foglalkozik.

A monitor működését alapvetően a monitor nyelv parancsai szabályozzák. E parancsok értelmezéséhez és feldolgozásához a monitor táblák nyújtanak segítséget. A monitor rendszer feladata a rendszer egy tetszőleges alrendszerének működtetése. Az alrendszer nyelven kapott utasításokat a monitor rendszer az illető alrendszer táblái alapján értelmezi és hajtja végre. Eközben használja saját, általános célú monitor rutinjait is az alrendszer speciális műszaki feladatot megoldó moduljai mellett. A monitor rendszer az alrendszer moduljait egy közös modulkönyvtárból veszi ki.

Egy alrendszer közös modulkönyvtárbeli moduljainak létrehozásáról természetesen előre kell gondoskodni. Erre szolgál az egységes rendszer igényének megfelelő programozó nyelv és annak fordító programja [7] .

Mint látható, a teljes rendszer különböző típusu alrendszerekből áll. Vannak mérnöki és vannak speciális alrendszerek. A mérnöki alrendszerek feladata közvetlen műszaki feladatok megoldása. A speciális alrendszerek feladata a műszaki feladatokat megoldó alrendszerek működtetésének előkészítése, a rendszer és az adatbázis kiszolgálása.



2. ábra

Az egységes gépipari tervező rendszer elvi felépítése

Az alrendszert generáló alrendszer állítja elő az összes többi alrendszer tábláit.

A kisegítő alrendszer tulajdonképpen a monitor bővitménye. Ez lehetővé teszi mindazon feladatok ellátását, amelyek szükségesek a szokásos operációs rendszer és kisegítő program funkciók közül pl. szövegszerkesztés, fájl manipulációk, programfuttatás stb.

Az adatbázist kezelő alrendszer arra szolgál, hogy előre létre lehessen hozni az alrendszerek működtetéséhez szükséges adatbázisokat. Feladata: ezek feltöltése és tartalmuk változtatása /módosítás, törlés, beszúrás, stb./. Erre az alrendszerre hárul ezeken felül minden, az adatbázissal kapcsolatos külső feladat elvégzése.

Az egységes gépipari rendszer felépítéséből következik, hogy a megoldandó feladatokat az alábbiak szerint célszerű csoportosítani:

- mérnöki alrendszerek
- bemenő nyelvek
- csatlakozási felületek
- adatbázis és adatbáziskezelő alrendszer
- programozó nyelv
- monitor rendszer és a többi speciális alrendszer

A feladatok alrendszerekre való bontása elsősorban a műszaki feladatoktól függ, ezért alapvetően technológusi, konstruktori feladat.

A bemenő nyelvek kialakítására - a mérnöki ismeretek mellett - a számítástechnika alkalmazásának lehetőségeit is jól ismerő szakemberekre van szükség.

A csatlakozási felületek kidolgozóinak - az alapos műszaki ismeretek mellett - jól tájékozottnak kell lenniük a monitor rendszer belső szolgáltatásaiban is.

Az adatbázis specifikálása elsősorban műszaki feladat, de az adatbázis és az adatbázist kezelő alrendszer létrehozása az adatbank témakörben különleges számítástechnikai ismereteket igényel.

A megfelelő transportábilis kisszámitógépes programozói nyelv kifejlesztése a fordítóprogramok készítésének témakörében speciális számítástechnikai ismereteket kíván.

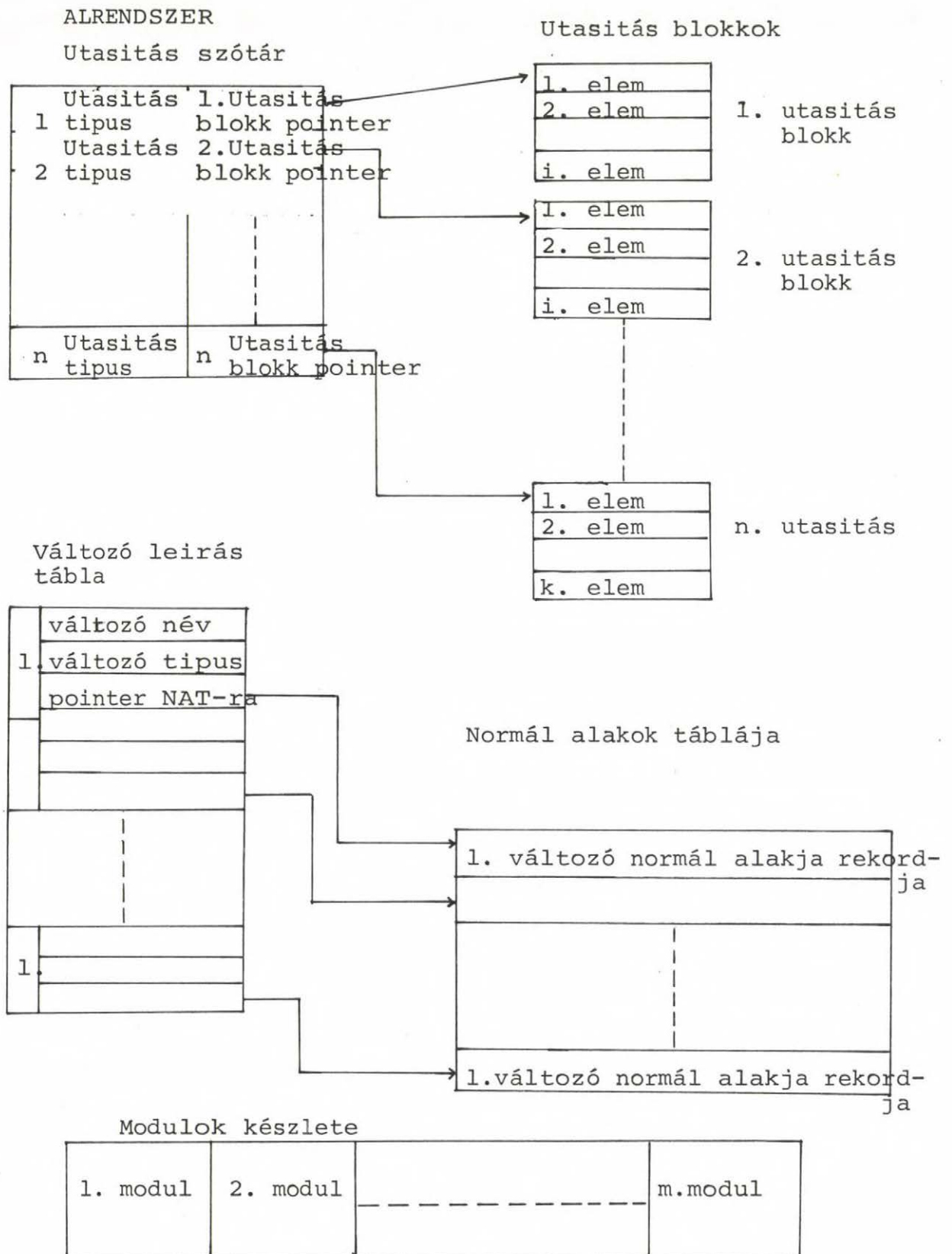
A monitor rendszer és a rendszer közvetlen alrendszereinek kifejlesztéséhez műszaki, rendszertechnikai és számítástechnikai ismeretekkel rendelkező kutatók szükségesek.

Mivel a teljes rendszer minden eleme csatlakozik a monitor rendszerhez, a rendszer kifejlesztése során a rendszerelemek kölcsönhatása érvényesül, ezért a munkákat párhuzamosan célszerű indítani.

2.5 Az alrendszerek felépítése

Az egyes alrendszerek, mint az a 2. ábrából is kitűnik, nem képeznek hagyományos módon összefüggő programot. Bár a rendszer lehetővé teszi - ezt a későbbiek során látni fogjuk -, hogy egy alrendszert úgy kezeljen, mint egy önálló programot, itt az alrendszerek e rendszerben szokásos felépítését tárgyaljuk.

Látszólag az alrendszerek táblák és modulok laza halmazából állnak, ahogy ez az alrendszer felépítés a 3. ábrából első közelítésében kitűnhet. A 3. ábrán a változó leírás és normál alakok táblájának kivételével azt látjuk, ahogy az al-



3. ábra

Alrendszer vázlatos felépítése

rendszer a háttértárolón elhelyezkedik. Vegyük sorra az alrendszer részeit.

Az alrendszer utasítás szótára tartalmazza belső formában mindazokat az utasításokat, amelyek az alrendszer bemenő nyelvén előfordulhatnak. Minden egyes utasítás tipushoz tartozik az utasítás szótárban egy mutató, amelynek alapján az utasításhoz tartozó utasításblokk megtalálható.

Az utasításblokk a rendszer egyik legfontosabb fogalma. Az utasításblokk sorai elemi utasításokat, úgynevezett utasítás elemeket, egyszerűbben elemeket tartalmaznak. Az utasításblokk minden sora egy ilyen elemet tartalmaz. Ezen elemek készlete nem tulságosan bő. Ezekkel az elemekkel lehet leírni azt, hogy milyen módon és milyen programokkal kell a felsimert bemenő nyelvi utasítást feldolgozni. Az utasításblokk minden egyes elemtipusa mögött egy monitor rendszerbeli speciális program áll, amely végrehajtja az elem által kitűzött feladatot a bemenő nyelvi utasítás egészén, vagy megfelelő részén.

Ezen elemek és a mögöttük álló elemfeldolgozó programok többek között előírhatják más monitor programok mozgósítását vagy méginkább a megfelelő modul vagy modulok megkívánt sorozatának mozgósítását. Az utasításblokk, mint a későbbiekben látni fogjuk, igen rugalmas eszközt nyújt az alrendszerre munkafolyamatainak vezérlésére.

Az utasításblokk tulajdonképpen egy olyan koordinátor, amely kapcsolatot teremt a bemenő nyelven megfogalmazott feladat és az azt végrehajtó modulok között. Ennek a koordinátornak az utasításait az utasításblokk elemei tartalmazzák, amelyeket a rendszer megért és végrehajt. Az utasításblokkokkal összeszervezett alrendszer részek a monitor rendszer közreműködése mellett már szoros rendszertechnikai egységet képeznek.

A változó leírás tábla tulajdonképpen egy tartalomjegyzék, amely háromszavas rekordokból áll. A bemenő nyelvi utasítások minden új változója kap egy ilyen rekordot. Esetleg a konstans értékek és névvel nem nevezett implicit változók tárolására is ezt a formát választjuk. A változó leírás tábla minden rekordja tartalmazza a változó nevét, típusát és egy mutatót arra a táblára, amely magának a változónak a jellemzőit tartalmazza. Ez az úgynevezett normál alakok táblája.

A normál alakok táblája rekordokból áll. Minden rekordja megfelel az illető típusu változó előre definiált kanonikus alakjának. E rekordok különböző hosszúságúak a változók típusától függően, de szigorúan megszabottak egy-egy változó típusra. E tábla rekordjait a változó leírás tábla alapján lehet elérni.

Például, ha egy névvel nevezett pont geometriai definíciójáról van szó a bemenő nyelvi utasításban, akkor a pont neve és típusjele a változó leírás táblabeli rekordjának első két szavába lesz letéve. A pontokat feldolgozó geometriai modul helyezteti majd el a normál alakra utaló mutatót a változó leírás táblába akkor, amikor ezen geometriai definíciót feldolgozta.

Egyidejűleg elhelyezteti az illető pont szabványos alakra hozott adatait a normál alakok táblájába, arra a helyre, ahova a mutató a változó leírás táblában mutat.

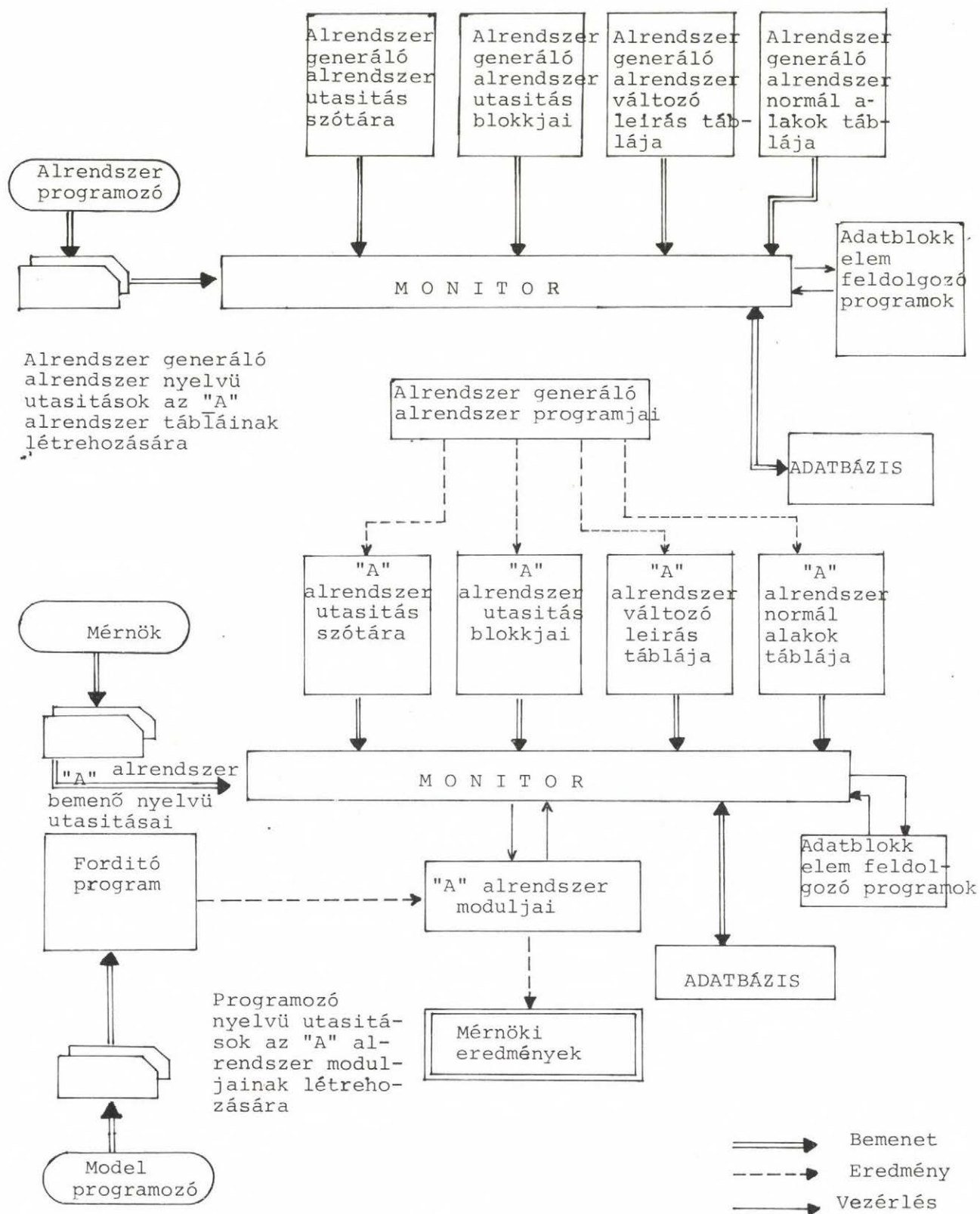
Látható, hogy ez az eljárás szinte teljesen megegyezik a KGST moduláris NC programozó rendszerben alkalmazott módszerrel [6,8]. Ugyanakkor teljesen különbözik az ICES és ISzP megoldásától [4,5]. E nagyszámítógépre korábban kidolgozott rendszerek az alrendszereken belüli adatforgalmat szinte kizárólag a FORTRAN programozó nyelv COMMON blokkjaira alapozták. Egyben viselték ennek a megoldásnak minden nehézségét és korlátját. E korlátok részleges leküzdésére a FORTRAN - nyelvet

kiterjesztették, megfejelték, ami a rendszerüket tovább bonyolította. Megoldásuk azonban elavultnak, nehézkesnek tűnik és semmiképpen nem alkalmas kisszámitógépes alkalmazásra.

A 4. ábrán szemléltetjük egy alrendszer, /nevezzük "A"-nak/ létrehozását és működtetését a monitor rendszer irányítása alatt. Külön nem jelöltük az adatbázis létrehozásának folyamatát, de megjegyezzük, hogy az adatbázis létrehozása az adatbáziskezelő alrendszer segítségével hasonló, a rendszeres használatot megelőző folyamat, mint az alrendszer generálás folyamata.

A 4. ábrán jól látható, hogy az "A" mérnöki feladatmegoldó alrendszer tábláit az alrendszert generáló alrendszer utasításai alapján az "A", és minden más alrendszer felépítéséhez hasonló alrendszert generáló alrendszer hozza létre. Magát az alrendszert generáló alrendszert, annak tábláit és programjait természetesen megelőzőleg kell létrehozni. Ez a rendszerfelépítő dolga és lényegében kézi feladat.

A mérnöki alrendszer létrehozásának másik fázisában, az "A" alrendszer moduljainak elkészítésében a monitor rendszer nem vesz okvetlenül részt. A program fejlesztésének ez a része szokásos operációs rendszerfeladat. Ezért célszerűbbnek látszik, hogy a kisszámitógép tulajdonságait figyelembe véve a programfejlesztés munkáját leválasszuk a monitor rendszer feladatairól. Így a modulépítés a szokásos módon megy végbe egy olyan programfejlesztő rendszerrel, amely kielégíti az egységes tervező rendszer igényeit. A monitor rendszer feladata lényegében futtatás, még akkor is, ha bizonyos önépítő tulajdonságokkal is rendelkezik a rendszer speciális alrendszerei révén. Természetesen a modulok programozására kifejlesztendő [7] programnyelv szoros kapcsolatban van a monitor rendszerrel.



4. ábra

"A" alrendszer létrehozásának és működtetésének elvi vázlata

Az "A" alrendszer futtatása a mérnök alkalmazó utasításai alapján indul meg. Az "A" alrendszer bemenő nyelvén megfogalmazott műszaki feladat feldolgozásában részt vesznek az "A" alrendszer táblái és moduljai. Az utasítás szótár alapján a monitor felismeri az utasítást és hozzárendeli a megfelelő utasítás blokkot. Az utasítás blokk elemei alapján a monitor behívja a megfelelő elem feldolgozó programokat. Ha a monitor modulmozgósító utasításblokk elemet talál, akkor az ennek megfelelő elemfeldolgozó program gondoskodik arról, hogy a monitor átadja a vezérlést a megfelelő "A" alrendszerbeli modulnak. A monitor addig dolgozik egy utasításblokkon, amíg annak végére nem ér, amely egyben egy "A" alrendszerbeli bemenő nyelvi utasítás feldolgozásának végét is jelenti. Ezután a monitor beolvassa a következő mérnöki utasítást. Ez volt az egyik alapvető működési mód. Később látni fogjuk, hogy a feldolgozás számos egyéb módja is lehetséges.

Bár a mérnöki eredményeket a modulok szolgáltatják, a monitor gondoskodik azok elhelyezéséről. Ezek kialakításában nagy szerepe van a csatlakozási felületek tervezőinek, mert az alrendszerek munkájának összehangolását elősorban itt lehet biztosítani.

Vessünk újra egy pillantást a 2. ábrán látható teljes rendszerre. Ha ezt összehasonlítjuk az ICES és ISZP célrendszerekkel [4,5], figyelembe véve a 3. és 4. ábrát is, azt látjuk, hogy kishátszámítógépes rendszerünkben elhagytuk az adatleíró alrendszert. E nagyszámítógépes rendszerekben az adatokat az alrendszerekben belüli COMMON mezőkben tárolják. Ahhoz, hogy az alrendszerek közötti információcserére lehetőséget tudjanak biztosítani, kénytelenek hatalmas mechanizmust létrehozni és mozgatni. Ez a mechanizmus átszövi a teljes rendszert, időt és értékes memóriaterületet igényel.

Már az alrendszer generáló alrendszer nyelvében is vannak olyan utasításai, amelyekkel byte pontossággal kell meghatározni egy-egy változó COMMON mezőbeli helyét az alrendszer COMMON-jában. Az ezekre vonatkozó tartalomjegyzéket ugyan a generáló alrendszer elkészíti, de az alrendszer COMMON mező megtervezése és kézbentartása az alrendszertervezőre súlyos feladatot ró, amely e mellett számos hibalehetőség forrása.

Van egy adatleíró alrendszer is, amelynek segítségével leírhatók az alrendszerek adatstruktúrái. Ez a speciális alrendszer elkészíti az adatleírások tábláját, amely szorosan hozzátartozik az alrendszerek rendszeresített tábláihoz. Az alrendszerek generálása így még egy lépcsőből áll. Az adatleíró alrendszert is alkalmazni kell a generáló alrendszer után. Az alrendszerek közötti adatátvitel programjai az adatstruktúra leírás alapján gyűjti ki az alrendszer COMMON-jából az adatokat, nagy apparátussal rendezgetik azokat, mielőtt a megjelölt helyre letelhették.

A mi rendszerünkben ez jóval egyszerűbben valósítható meg. A KGST rendszerben bevált módon az adatainkat egy előre definiált táblában, a normál alakok táblájában helyezzük el. Ezeket felkeresni, kigyűjteni akár név, akár típus szerint a változó leírás tábla alapján sokkal egyszerűbb. Egy következő alrendszernek akár az egész korábbi normál alakok táblája, akár ennek egy meghatározott része átadható a hozzá tartozó változó leírás táblával együtt. Ehhez jóval egyszerűbb mechanizmus, idő és memóriaterület szükséges, ami kishátszámítógépes alkalmazás esetén különösen nagy előny. Megjegyezzük, hogy a KGST rendszert már eleve úgy terveztük, hogy kishátszámítógépes alkalmazásra is megfelelő legyen.

Egy alrendszernek a háttértárolón azokon kívül is lehetnek fájlljai, amelyeket egységesen minden alrendszerhez előre hozzárendeltünk. Ezek lehetnek ideiglenesek, amelyek csak az al-

rendszer működése alatt érvényesek, és lehetnek olyan fájlok, amelyek megőrződnek az alrendszer futása után is. E fájlokkal tetszőleges adatátadás valósítható meg az alrendszerek, de még a modulok között is.

A fájlok lekötését kérhetik a modulok programból, a megfelelő monitor rendszerbeli rutinok meghívásával. A feladatot e rendszerrutinok hajtják végre a hívási paraméterek adatai szerint.

Méginkább lehetséges e fájlok külső definiálása az alrendszer megindítása előtt. Arra is lehetőség van, hogy a fájl manipulációkat az alrendszer futása alatt írjuk elő külső parancsokkal. Mindezekre eszközöket biztosít a monitor nyelv és az utasításblokkok megfelelő elemei, amelyek végül is ugyanezeket a rendszerrutinokat aktivizálják a fájl manipulációkhoz, mint amelyeket a modulok hívnak a programból. Így a mérnöknek lehetősége van eredményeit, részeredményeit olyan helyekre gyűjteni, amelyeknek saját maga adhat neveket és amelyeket később felhasználhat.

2.6 Bemenő nyelvek

A bemenő nyelvek különböző feladatok megfogalmazására szolgálnak. Ezekkel írjuk le azokat a problémákat, amelyek megoldását a számítógépes programoktól várjuk. Más megfogalmazásban, ezekkel állítjuk fel a megoldandó feladatok nyelvi modelljét.

Ezen nyelvekre vonatkozó alapvető követelményeket részletesen taglalja az [1] tanulmány. Ezért azt a kérdést itt csak érintjük. Legfontosabb követelmény, hogy a nyelv közel álljon a mérnöki gondolkozáshoz, és alkalmas legyen a kisszámítógépen való feldolgozásra is. Ezért a bemenő nyelven megfogalmazott program rendezett mondatok sorozata. A mondatokat olyan szabályokkal és olyan formátumban kell írni, amelyek egyértelmű

kifejezéseket, nyelvtani szabályokat, jeleket és jelölési módokat alkalmaznak úgy, hogy a tévesztés lehetősége minimális legyen. Ezzel együtt jár, hogy a programozás módja könnyen elszámítható, a program pedig jól áttekinthető legyen. A szubjektív hatások kiküszöbölésére a bemenő nyelveket hibafelderítő rendszerrel kell kiegészíteni.

Fenti követelményeknek a [6] javaslatban specifikált KGST egységes moduláris NC programozási nyelv jó közelítéssel eleget tesz, ezért célszerű, hogy az egységes kishámitógépes tervezőrendszer bemenő nyelvének is ez szolgáljon modelljéül.

Nem felesleges ismételtten hangsúlyozni, hogy a teljes rendszer egységes bemenő nyelvet használ. Láttuk, hogy minden egyes alrendszernek saját külön bemenő nyelve van. Ez azonban nem okvetlenül mond ellent annak, hogy a teljes rendszer egységes nyelvet használjon. Alapvető a nyelv szerkezetének, nyelvtani szabályainak, jeleinek és jelölési módjainak egységessége. A különböző alrendszerek a különböző fogalmakat más-más szavakkal, de azonos módon jelölik és a feladatokat is azonos módszerekkel írják le.

Különösen fontos, hogy a teljes rendszer speciális feladatait megoldó alrendszerek nyelvei is hasonlatosak legyenek a műszaki feladatokat megoldó alrendszerek nyelvéhez. Ez a rendszer egységességének, együttes kezelésének alapvető feltétele.

A bemenő nyelveket rendeltetésük szerint az alábbi csoportokra oszthatjuk:

- műszaki feladatlíró bemenő nyelvek,
- mérnöki rendszerkezelő nyelvek,
- alrendszer generáló nyelv,
- adatbázis kezelő nyelv.

Ezeket itt külön-külön nem taglaljuk, egy részletes rendszertervben majd mindegyiküket specifikálni kell, az egységes nyelv szerkezetének szigorú lerögzítése mellett. Kezdetben elengedhetetlennek látszik a rendszer speciális nyelvei alapszókincsének kialakítása, különösen az alrendszer generáló nyelv és a rendszerkezelő nyelvek létrehozása. Ha ezek léteznek, a rendszer önépítő, önfejlesztő lehetőségei már kihasználhatók a műszaki jellegű bemenő nyelvek terén. A rendszer bemenő nyelvei specifikálásának modelljéül szolgálhat az a módszer, amelyet a KGST egységes NC programozó rendszer bemenő nyelvének specifikálásánál követtünk. Ott a bemenő nyelvi mondatok szintaktikai és szemantikai leírása mellett rögtön rögzítettük a várható eredmény normál alaktáblabeli rekordjának felépítését, belső kódokat adtunk külön a fogalmaknak és külön a módosító szavaknak, ezen kívül meghatároztuk mindjárt a feldolgozást végző modulra való utalást is. Ezt az utat itt értelemszerűen kell követni a rendszer felépítési jellegének, az eltérő szerkezetnek a figyelembevételével.

Az 1. ábrán a rendszernek adott külső utasítást parancsnak neveztük. Tettük ezt azért, hogy a bemenő nyelvi utasításokat egységesen jelöljük. A bemenő nyelvi mondatok két alapvető típusát különböztetjük meg. Az egyik típust nevezzük utasításnak. Ez az általánosabb, minden egyes alrendszer ilyen bemenő nyelvi utasításokkal dolgozik. Mindaz, amit eddig elmondtunk, ezekre az utasításokra vonatkozik. Ezen utasítások feldolgozása a rendszerben egységes módon megy végbe.

Szükséges azonban megkülönböztetni olyan parancsokat, amelyeket azonnal kell teljes egészében végrehajtani. Függetlenül attól, hogy a rendszer éppen milyen állapotban van, hogy a feldolgozások menete hol tart, ezekre a különleges parancsokra a rendszernek azonnal reagálnia kell. Ezt a mondattípust nevezzük direktívának.

A direktiva olyan speciális utasítás, amely elsőbbséggel rendelkezik minden más utasítással szemben. Ha rendszer ilyen-nél találkozunk, akkor munkáját félbeszakítva a direktívát azonnal feldolgozza. Ezért a direktívákat meg kell különböztetni a szokásos utasításoktól. Azonban az is elengedhetetlen, hogy a direktívák eleget tegyenek az egységes bemenő nyelvre szóló követelményeknek. Ennél fogva a direktívák csaknem teljesen megegyeznek az utasításokkal. Az egyetlen különbség, hogy a direktiva speciális karakterrel kezdődik. Más utasítás ilyen karakterrel nem kezdődhet. Hogy milyen kezdő jelet, esetleg jeleket célszerű a direktiva jelölésére használni, az külön megvitatandó kérdés.

Olyan jelre van szükség, amely a kisszámitógépek nyomtató és kijelző készülékei többségének karakterkészletében megtalálható, és elég feltűnő a rendszer alkalmazójának a számára.

A direktívákat a monitor nyelvbe csoportosítottuk. A monitor az alrendszerekhez hasonlóan rendelkezik direktiva szótárral és ezeknek megfelelő direktiva blokkokkal. Az egyetlen különbség, hogy a monitor rendszer a direktiva szótárat állandóan kéznél tartja. Ez a felépítés mindjárt érthető, ha meggondoljuk, hogy a monitor rendszer ezekkel a direktívákkal hív be újabb és újabb alrendszereket, függeszti fel, menti ki, szakitja félbe, törli vagy kezdeményezi az alrendszerek működését. A direktívák közé tartoznak azok a mondattípusok is, amelyekkel az alrendszerek közötti adatforgalmat lehet előre kijelölni. Ezért célszerű, hogy a direktívák körébe tartozzanak az alrendszerekre vonatkozó fájlkezelő utasítások is. Tehát a direktívák olyan utasítások, amelyek előnyt élveznek, minden alrendszerhez szükségesek lehetnek, külön kezelésben részesülnek, emiatt speciális jellel tüntetjük ki őket.

A kisegítő alrendszerre azért is van szükség, hogy azokat a rendszerfeladatokat, amelyek nincsenek közvetlen kapcsolatban az alrendszerekkel - tehát amelyeket az alrendszerek mű-

ködetetése előtt vagy után lehet végrehajtani - azokat ne kelljen a direktívák közé sorolni. Ez túlságosan megterhelné a direktíva szótárat, csökkentené a monitor rendszer rugalmasságát, bővithetőségét. Így a szövegszerkesztési, általános fájlkezelési, másolási stb. feladatok megoldására, az alrendszerek bemenetének előkészítésére, az alrendszerek eredményeinek rendezésére, stb. a kisegítő alrendszer utasításai szolgálnak. Megjegyezzük, hogy a kisegítő alrendszert is a mérnöki alkalmazó használja. Tehát utasításainak csakugy mint a direktíváknak is a műszaki feladat leíró nyelvi részhalmazzal együtt a mérnök számára kézenfekvőnek kell lennie.

Szükséges még rámutatni a bemenő nyelvek, a feladatok felbontása, az alrendszerek szótára és a kisszámitógépes alkalmazás közötti összefüggésre.

A feladatok lebontásában első szinten állnak az alrendszerek. A rendszerben meg van az a lehetőség, hogy bármilyen kis utasításcsoportot, esetleg egyetlen bemenő nyelvi utasítással rendelkező egyetlen modulból álló egységet is alrendszernek tekintsen, ha van ezen alrendszernek neve a direktíva szótárban és rendelkezik a 3. ábra szerinti felépítéssel, vagyis, ha az alrendszer generáló alrendszer hozta létre. Elképzelhető tehát egy tágabb feladatot megoldó alrendszer felbontása alacsonyabb szintű alrendszerekre, ha sikerül ezen alacsonyabb szintű alrendszerek sorozatának munkáját összehangolni, úgy, mintha egyetlen nagyobb alrendszer dolgozott volna. Ekkor az első szintű alrendszer bemenő nyelvű utasításkészletét is fel lehet osztani kisebb halmazokra az alacsonyabb szintű alrendszerekre való tagolás szerint. Ezen módon alaposan csökkenthetők azon táblák méretei, amelyekkel a lebontott alrendszerek rendelkeznek. Lecsökken az utasításszótár mérete, meggyorsulhat és egyszerűsödik a keresés művelete. Egyszóval kisszámitógépre hatásosabban alkalmazható szerkezet alakítható ki.

Ugy tűnik hasonló hatás érhető el azzal is, ha meghagyjuk ugyan az első szintű alrendszerekre való tagolást, de a tel-

jes feladatot mégis sorrendben következő feladatcsoportokra bontjuk. Az alrendszer bemenő nyelvű utasításait ennek megfelelően részhalmazokra osztjuk. Az alrendszer utasításszótárába pedig mindig az aktuális, a kívánt feladatcsoportnak megfelelő részhalmazt töltetjük be külön utasítással. Ha a feladatcsoportok, azaz szótárak nem tulságosan gyakran váltják egymást, azaz a teljes feladatot fel lehet elkülönülő szakaszokra bontani, e módszer hatása hasonlóan kedvező lehet, mint az alacsonyabb szintű alrendszerekre bontás.

A rendszerben célszerű mindkét megoldást lehetővé tenni a harmadikkal együtt, amely az eredeti feladatlebontás nélküli eset. A követendő elv ebben is , mint az egész rendszerben, az, hogy bizonyos előnyökért csak az fizessen, aki azt élvezni akarja. Tehát a megoldási variációk ne terheljék a teljes rendszert, ne okozzanak nagyobb rezidens méretű monitort a szükségesnél.

Külön említést érdemel az alrendszer generáló alrendszer bemenő nyelve. Hangsúlyozzuk, hogy ez is hasonló szerkezetű mint a többi, tehát az egységes nyelv része. Ennek utasításai hatására töltődik fel a generálandó alrendszer utasítás szótára és töltődnek fel ezen utasítás feldolgozását irányító utasításblokkok belső utasítás elemekkel.

2.7 Csatlakozási felületek

Egységes rendszer nem alakítható ki anélkül, hogy ne lennének szigorúan rögzítve, szabványosítva a közbülső adatforgalom módszerei, formátumai. Egységesíteni kell az alrendszerek közötti adatforgalmat csakugy, mint az alrendszereken belülieket a modulok és modul csoportok között.

Iránymutató kísérlet egy technológiai tervező rendszerre nézve a [10] tanulmány, valamint a KGST moduláris NC programozási rendszerében alkalmazott eljárás.

Mint láthattuk az egységes kishszámítógépes rendszer több lehetőséget, eszközt is nyújt a csatlakozási felületek tervezőinek és az alrendszer programozóknak arra, hogy kialakítsák a csatlakozási felületeket. Alapvető elv, hogy a rendszerben az adatátadás fájlokon keresztül megy végbe.

Csatlakozási felületek kialakítására rendelkezésre áll az alrendszerek normál alak táblája, az alrendszerekhez és modulokhoz tetszőlegesen köthető fájlok, valamint az adatbázis.

Egyszerűbb esetben az alrendszerek és az alrendszereken belül a modulok közötti adatátadásra elegendő a normál alakok táblája. Ezek monitor didektívákkal is átrendezhetők egy következő alrendszer számára vagy kívánságra megőrizhetők tetszőleges időpontban való felhasználásig.

Fájlok definiálásával és megfelelő formátumu kitöltésével lehet olyan csatlakozási felületeket képezni, mint amilyen az NC programozásban a CLDATA fájl. Ez a lehetőség szinte megszorítás nélküli. Azonban szükséges hozzá elkészíteni azokat a modulokat, amelyek ezen fájlok kitöltését és leolvasását megoldják.

Az adatbázis felhasználása csatlakozási felület képzésére csak akkor célszerű, ha az egyik alrendszer eredményei közhasznúak és érdemes azokat letenni a közös adatbázisba. Külön meggondolandó, hogy az adatbázisnak legyen-e egy ideiglenesebb jellegű része, amely csatlakozási felület képzésére is alkalmas. Az is meggondolandó, hogy lehet-e és érdemes-e az adatbázis mechanizmusát e célra felhasználni.

2.8 Adatbázis

Korszerű elveken alapuló adatbázis tényleges létrehozása kishszámítógépre egyike a legsúlyosabb feladatoknak. Az általános követelményeket [1]-ben megfogalmaztuk. A megoldás módjá-

ra iránymutató a [6]-ban specifikált módszer. A kisszámitógépes megvalósítás útjainak feltárását és a megvalósítás lehetőségeit a [11] taglalja. Feltétlenül kívánatos külön tanulmányt készíttetni és kísérleteket végezni az egységes kisszámitógépes tervezőrendszer igényeinek megfelelő adatbázis létrehozására.

Ugy tűnik szükséges lehet a kisszámitógépes alkalmazás figyelembevételével az egyes tervezői alrendszerek számára külön rész adatbázis létrehozására egy nagyobb, esetleg a teljes rendszerre érvényes adatbázisból. Ez teremtené meg a kapcsolatot a külvilág felhalmozott tapasztalatai és az alrendszer között.

Az is elfogadható megoldás, ha e rész adatbázist nagyszámítógépen állítják elő. Ekkor a teljes rendszer adatbázisa nagyszámítógépre tehető ott kezelhető és a kisszámitógépes rendszer mindig csak az előre létrehozott rész adatbázissal foglalkozik.

A rendszerben az adatbázishoz kapcsolódik az adatbázis kezelő alrendszer. Ezt úgy kell kialakítani, hogy az esetleges részadatbázisokkal tudjon manipulálni. Ehhez minden bizonnyal az szükséges, hogy ezen részadatbázisok felépítése eléggé egységes legyen az egységes kezelhetőséghez.

Akár rész, akár teljes adatbázis megvalósítása lehetséges, alapvető igény, hogy az adatbázis kezelése a rendszer egyéb alrendszereihez hasonló módon és olyan nyelvvel legyen lehetséges, amelyet a felhasználó a többi alrendszerénél megszokott.

A modulok és az adatbázis kapcsolatát a rendszerben központi rutinok végzik, hasonlóan a KGST moduláris NC programozó rendszerében alkalmazott eljárással.

2.9 A programozó nyelv

A programozó nyelvről külön tanulmány készül [7] . A vele szemben támasztott általános követelményeket 1.3-ban tárgyaltuk. Itt rá kívánunk mutatni a programozó nyelv szoros kapcsolatára a teljes rendszerrel, különös tekintettel a monitor rendszer belső kialakítására.

Itt elsősorban nem arra gondolunk, hogy magának a monitor rendszernek jórészt is ezen nyelven írhatjuk meg. Másról van szó.

Az alrendszerek moduljait a programnyelven írják. Ahhoz, hogy a monitor rendszer a kisszámítógép erőforrásait lehetőleg kedvezően tudja mozgósítani és kezelni a modulok futása alatt, szükséges, hogy a programnyelv tartalmazzon olyan utasításokat, amelyek a dinamikus program és adatkezeléshez szükségesek. Ezzel tehermentesítse a modulprogramozót attól, hogy ő figyelje állandóan, programjai és adatai hogyan férnek el az operatív memóriában. Röviden kapcsolatot kell e programnyelvnek létesítenie a monitor rendszer [9] -ben kipróbált módszert alkalmazó részével.

Ez azt jelenti, hogy a részletes rendszertervek során fokozatosan közelíteni kell a programnyelv lehetőségeit az egy-egy tervező rendszer igényeihez és fordítva is.

A korábbi nagyszámítógépes rendszerek például [4,5] úgy kerülték meg az alkalmazott FORTRAN nyelv és az IBM 360 DOS szerű operációs rendszerek korlátait, hogy megfejelték a FORTRAN-t és létrehoztak egy "precompiler"-t, amely a bővített FORTRAN nyelvű programot először lefordítja normál FORTRAN-ra beépítve egy sereg assembler nyelvű rendszer rutin hívást. A tényleges fordítás csak ezután következik. Nem az előkészítés ilyen bonyolítása a zavaró, hanem az, hogy egy terjedős mechanizmus szükséges a feladat távolról sem optimális megoldásának

megközelítésére. Ugyanakkor rendkívüli mértékben függ az éppen alkalmazott operációs rendszer korlátaiktól is, és még így is csak az adatkezeléssel kapcsolatos dinamikus memória elhelyezést sikerül megközelíteni külön alrendszer és az egyes alrendszerek további bővítése árán.

Programok terén csak az IBM-nél szokásos azon korlátot tudták trükkökkel, idő és memória áldozattal leküzdeni, amely megtiltja különböző egyszer már összekapcsolt szubrutin csoportok u.n. load modulok közötti szubrutin hívást.

Itt ebben a rendszerben többről van szó. Általánosan elterjedt és szabványosított, MSZR kisméretű, kötelező erejű nyelv még nincs. Nem kell ragaszkodni meglévő korlátozásokhoz. Ezért olyan nyelvet kell létrehozni, amely lehetőleg mindenben kielégíti a tervező rendszer igényeit.

Ezért is meggondolandó, hogyan lehetne és célszerű-e a program nyelvbe beépíteni a korszerű adatbázisok egyik elemét az adat manipuláló nyelvet.

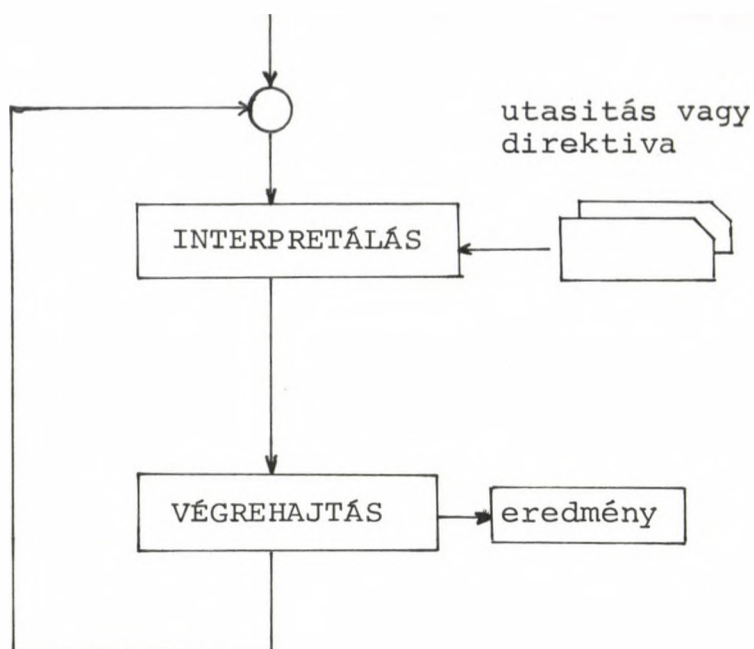
2.10 A monitor felépítése

Ebben a fejezetben a monitor vázlatos elvi felépítésének elképzelésével foglalkozunk, részletesen a monitort a 3. fejezetben tárgyaljuk.

A monitor alapvető feladata mint ez az 5. ábra egyszerűsített folyamatábráján látható a bemenő nyelvi utasítások értelmezése és feldolgoztatása.

A monitor munkáját direktívák irányítják. Ezek határozzák meg a bemenő nyelvi utasítások feldolgozásának mikéntjét, azt, hogy honnan vegye az utasításokat és hogy milyen alrendszer szolgáltassa ki. A monitor tulajdonképpen szervező, amely különböző információk alapján egy alrendszer munkáját irányít

ja.



5. ábra

A monitor elvi működése

A 6. ábra már részletesebb elvi vázlatot mutat. A monitor munkája a monitor rendszer behívásával kezdődik. Az inicializáló egység kezdi a tényleges munkát, főleg azzal, hogy leköti a szükséges fájlokat és betölti a direktiva szótárt.

A beolvasó egység lexikai táblái segítségével belső alakra hozza a beolvasott utasítást vagy direktívát. A direktívát első karakteréről felismeri és biztosítja különleges bánásmódját. A beolvasó egység lexikai táblái cserélhetők, így nagy rugalmasságot biztosít a karakterkódok, speciális elválasztójelek és többkarakteres szeparátorok cserélhetőségére. A beolvasó egység kimenete a beolvasott utasítás vagy direktiva belső alakja, amelyet tovább az interpretáló egységnek. Az interpretáló egység ezt tovább módosítja és olyan alakra hozza, amelyet a modulok vagy rendszer programok már fel tudnak dolgozni. Ehhez bármely modul vagy rendszer program hozzáférhet, de

az ábrán külön nem jelöltük az összekötő vonalak zavaró hatása miatt. Így is csak jelképesen helyeztük a folyamatábrába, tulajdonképpen nem oda való helyre. A továbbiakban a fenti folyamatot adottnak vesszük.

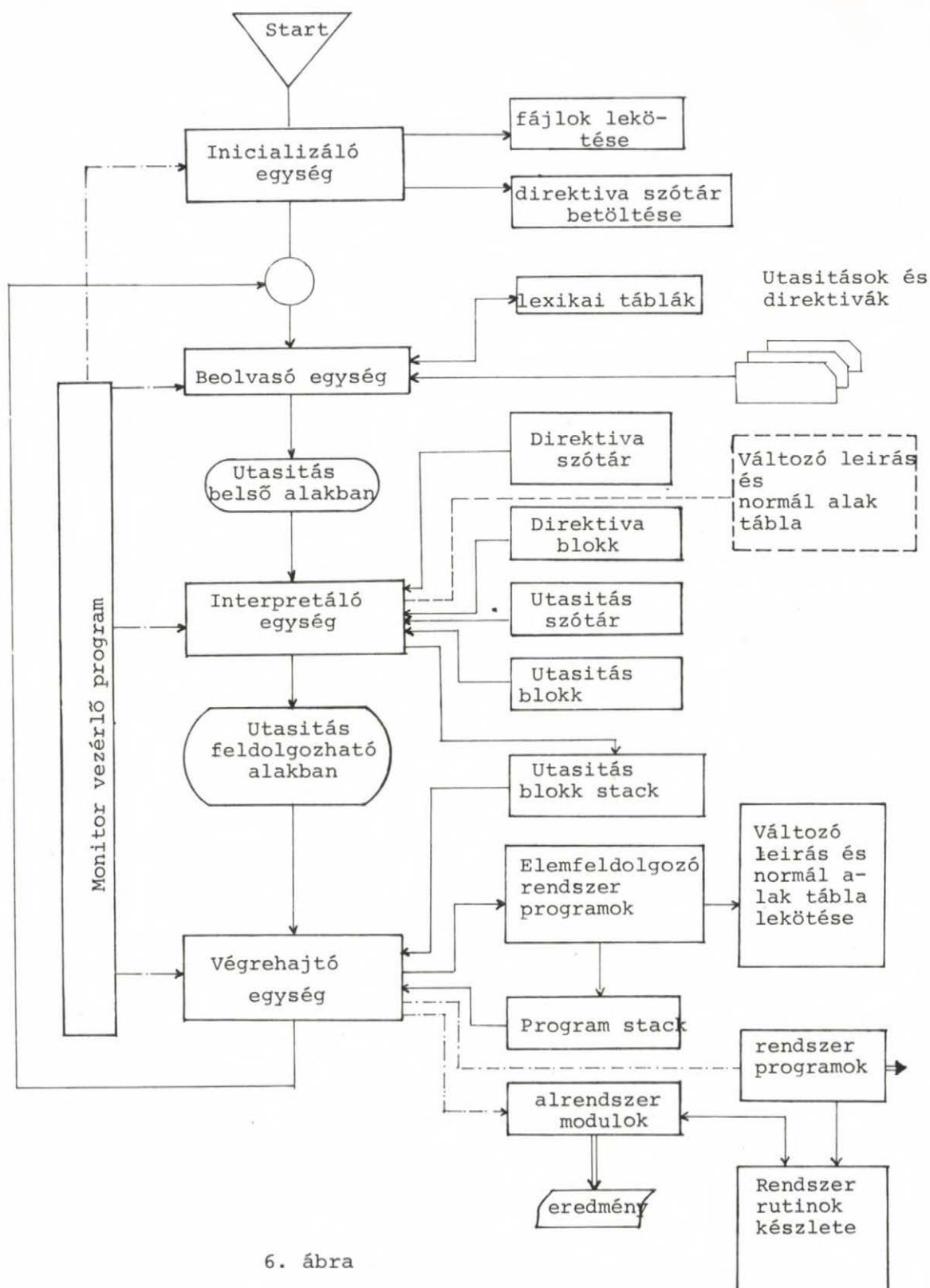
Egyszerűsítve az első bejövő parancs egy direktiva, amely egy alrendszer megnevezése. Ezt az interpretáló egység összehasonlítja a direktiva szótár tényezőivel. Ha nem talál egyezést, hibát jelez. Ha talált egyezést, felkeresi a direktiva blokkját. Ezt beteszi az utasítás blokk stackbe.

A monitor vezérlő programja ezután a végrehajtó egységet hívja meg. Ez az utasítás blokk stack sorait egymás után sorra veszi. Leveszi az első sort, meghívja az ennek megfelelő elem feldolgozó programot. Ez elvégzi a feladatát. Ha olyan elemet talál, amely a program stackbe töltet, akkor a megfelelő elemfeldolgozó rendszerprogram beteszi a program stackbe a megnevezett rendszer programot vagy rendszer programok egy sorozatát.

A végrehajtó egység állandóan vizsgálja, van-e megnevezett program a program stackben. Ha talál, mint esetünkben, akkor ezt a rendszer programot megindítja. Az alrendszer megnevező direktiva feldolgozását végző soron végrehajtó rendszer programok eredményül az utasítás szótárba beteszik a megnevezett alrendszer utasítás szótárát, kézbe veszik az alrendszer utasítás blokk tábláját és modul készletét. Ezenkívül inicializálják az alrendszer változó leírás és normál alak tábláját.

Ezután a végrehajtó egység visszaadja a vezérlést a monitor vezérlő programjának, amely újra behívja az interpretáló egységet. Ez leveszi a következő parancsot. Ez már lehet utasítás vagy direktiva.

Ha utasítás, akkor a betöltött alrendszernek megfelelően aktualizált utasítás szótárban keres egyezést, miután a bemenő



6. ábra

A monitor elvi vázlata

nyelvi utasítást belső alakra hozta. Ha nincs egyezés, akkor hibát jelez. Ha talál tényezőt az utasítás szótárban, amely egyezik a beolvasottal, akkor behozza utasítás blokkját és leteszi az utasítás blokk stackbe. Esetleg az interpretáló egység elvégzi az alrendszer változó leírás táblájának és normál alak táblájának automatikus kitöltését is egyszerűbb esetekben, például aritmetikai változókra. Ez opcionális lehetőség.

Ezután a vezérlést a végrehajtó egységhez teszi a vezérlő program. A végrehajtó egység az utasítás blokk stack beli utasítás elemek szerint megkezdí az elem feldolgozó rendszerprogramok meghívását. Ezek elvégzik feladatukat, például az alrendszer változó leírás táblájában lekötnek egy rekordot egy névvel nevezett változó számára. Megjegyezzük, hogy esetleg ezt a műveletet az interpretáló egység automatikusan is elvégezheti. Ha olyan elemet talál, amely programokat mozgósít, akkor a megnevezett programok nevét az adott sorrendben lehelyezi a program stackbe. E programok között lehetnek rendszer programok is, de méginkább az alrendszer modulneveire lesz itt hivatkozás .

A végrehajtó egység minden egyes újabb utasítás blokk stack beli sor lekérdezése előtt megvizsgálja van-e tényező a program stackban.

Esetünkben talál és most már meghívja az alrendszer megnevezett modulját. A modul feladata végrehajtása során szabadon használhatja a monitor rendszer rutinjait. Ilyen rutinok segítségével fogja például eredményét lehelyeztetni az alrendszer normál alakok táblájába, amikor ezzel egyidejűleg a változó leírás táblabeli vonatkozó rekord harmadik szavába is bekerül ez a mutató, amely jelzi hol van a vonatkozó változó-táblabeli rekord.

Modul ill. rendszerprogram hívások következnek mindaddig, amíg a programstack ki nem ürül. Ha kiürült, akkor a végrehajtó egység leveszi a következő elemet az utasítás blokk stack-

ből. Ha mind a két stack kiürült, a vezérlés újra visszakerül és az interpretáló beolvassa a következő utasítást.

Ez a folyamat mindaddig tart, amig direktívával meg nem szakítjuk.

A mérnök munkája így már világos. Egyetlen speciális utasítással, amely az operációs rendszernek vagy egy betöltő programnak szól, behozza a rendszert. Monitor direktívákkal kijelöli a kívánt input perifériát, választ esetleg output perifériákat is. Más direktívákkal a feldolgozás választható módjai közül kijelöli a megfelelőt. Ezután megnevezi a feladat végrehajtására alkalmas alrendszert.

Megfogalmazza feladatát ennek az alrendszernek a nyelvén /ezt már előre is megtehetette/ és beadja utasításait. Dolgozhat interaktívan, ekkor a rendszer soronként várja az utasításokat, de beadhatja egy csomagban is.

Az utasítások közben direktívákat is közölhet a rendszerrel. Ha más alrendszert nevez meg direktívájával, az előző automatikusan lezáródik és törlődik a monitor tábláiból. Megőrzésre csak azok a fájlok kerülnek, amelyeket előre kijelölt.

3. A MONITOR RENDSZERVÁZLATA ÉS MŰKÖDÉSE

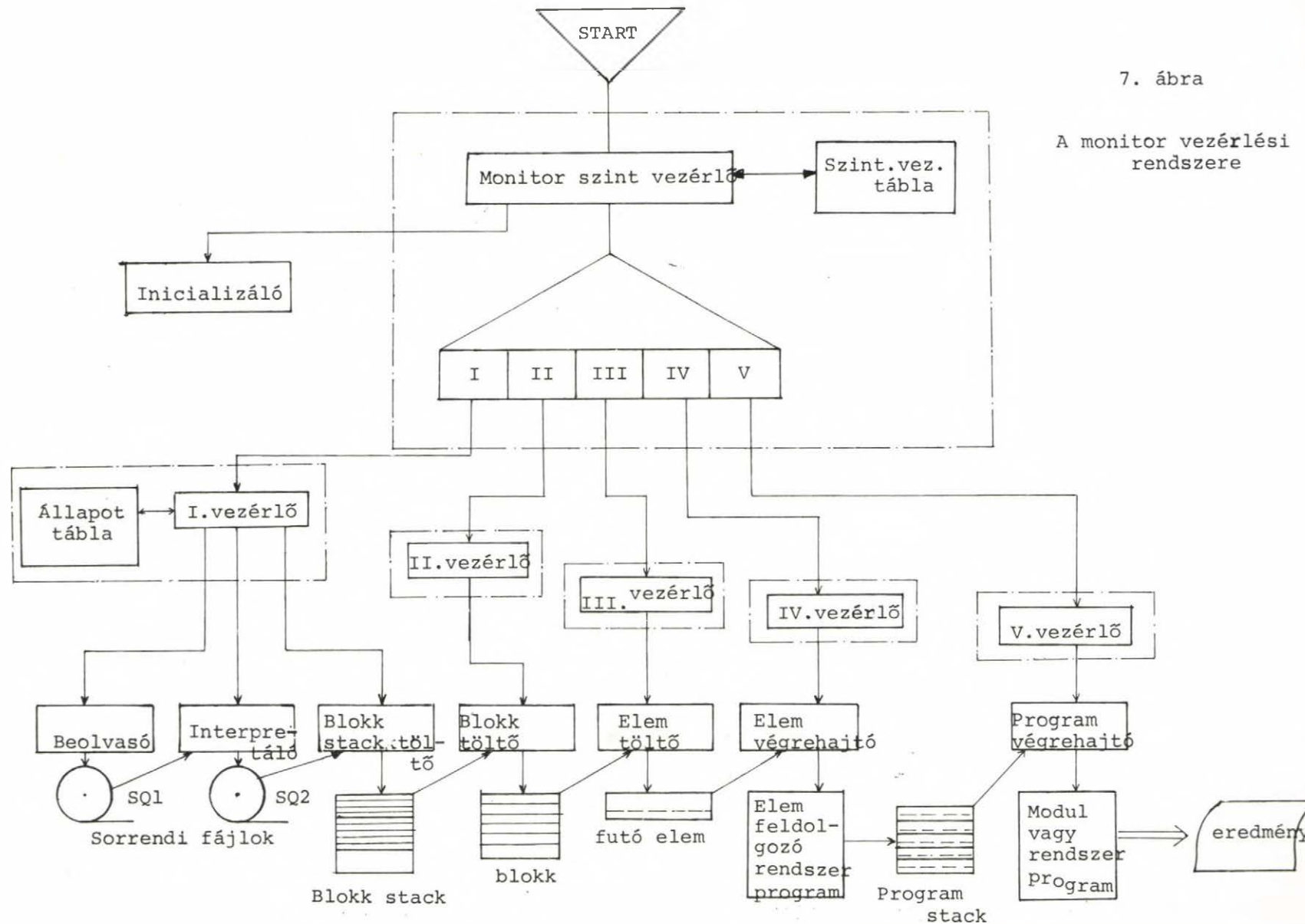
Az előző fejezetben /2.10/ megismerkedtünk a monitor elvi felépítésével és működésének alapjaival. Ebben a fejezetben további részletekre térünk ki. Nézzük a 7. ábrát.

Amint láttuk a monitor rendszer alapvető feladata, problémaorientált nyelven irt utasítások beolvasása, értelmezése és feldolgozásának irányítása. Magának a folyamatnak a kívülről való befolyásolására a monitor direktívák adnak lehetőséget, amelyek feldolgozása hasonló, és ugyanazt a mechanizmust használja, mint az utasításoké. Az alapműködés inkrementális, tehát sorról sorra valósul meg a feldolgozás. Míg egy mondatnál nem végzett, a következőt nem olvassa be. Ez az alapműködés azonban direktívákkal befolyásolható és a működési módok egy választéka áll majd a felhasználók rendelkezésére. A működési módot kijelölő direktívák hatására kitöltődik a monitor állapot táblája /lásd további ábrákon/. Ennek az állapot táblának megfelelő rovatai vannak arra nézve, hogy a monitor különböző egységei hogyan dolgozzanak. Például, hogy a beolvasó egység, az interpretáló egység mondatonként dolgozzon, azaz alapműködésben, vagy köteget üzem módban egy egész vagy részfeladat összes mondatát egyszerre dolgozza fel. Ugyancsak az állapot tábla megfelelő rovataiba kerülnek a perifériák kijelölésére vonatkozó direktívák előírásai. Számos más célra is használható ez a tábla. Például az alrendszerhez rendelt fájlok nevei, az ideiglenes alrendszer táblák és fájlok kimentésére vonatkozó előírások szintén ide rakhatók le.

Természetesen abban az esetben, ha köteget működésmódról van szó, gondoskodni kell az egymás után beolvasott, majd az egymás után interpretált utasítások végrehajtás előtti megőrzéséről. Ehhez sorrendi felépítésű fájlra van szükség a beolvasó egység után is és az interpretáló egység után is. Lehetőség van arra is esetleg, hogy csak a beolvasó egység dolgozzon csoportos módon. Ekkor az interpretáló egység az első

7. ábra

A monitor vezérlési rendszere



sorrendi fájlról mondatról mondatra szedi le a belső alakra hozott utasításokat és adja át a már ismert módon a végrehajtó egységeknek a feladatot. Másik eset, ha a beolvasó és interpretáló egység együtt dolgozik kötegelte üzem módban. Ekkor a második sorrendi fájl fog tölteni. A végrehajtó egység erről fogja levenni és átadni az információt a moduloknak, de itt már természetesen sorról sorra. Mivel magát a végrehajtó egységet az utasításblokk irányítja az illető utasításmondat feldolgozása során, az utasításblokk stack töltésének feladatát le kellett választani a kötegelve is üzemeltethető interpretálásról.

Fentiek után már nem jelent nehézséget, ha a beolvasó egység is előbb végig megy az utasítás kötegen és csak azután adja át az első sorrendi fájlt az interpretáló egységnek, amely ezután előállítja a második sorrendi fájlt.

A tárgyalt működési módokon kívül talán még más is elképzelhető. Azonban mindezekhez biztosítani kell a képletes kezdő és végzőjelet az utasításfolyamban. Ilyen kezdő és végzőjel lehet direktiva is, de akkor óvatosan kell bánnunk velük, mert azonnal végrehajtnak. Elképzelhető a kötegek kezdetét és végét olyan alrendszerbeli utasításokkal nevezni, amelyek minden alrendszerre közősek. Mint a rendszer egyéb helyein, itt is több megoldás lehetősége közül lehet választani.

Természetesen, ha az alpműködés valósul meg, akkor a SQL és SQ2 fájllokba csak egy-egy utasításnak megfelelő információ kerül. Ezek helye pedig úgy is rendelkezésre áll belül, mint az említett fájllok monitorbeli bufferei.

3.1 A monitor vezérlési rendszere

A 7. ábrán látható a monitor vezérlési rendszere a monitor alatt futó monitor egységek munkafolyamatának vázlatos jelölésével. Ez a 6. ábra folyamat diagramjának a vezérlés oldaláról

való megközelítése néhány feladat lebontásával kiegészítve.

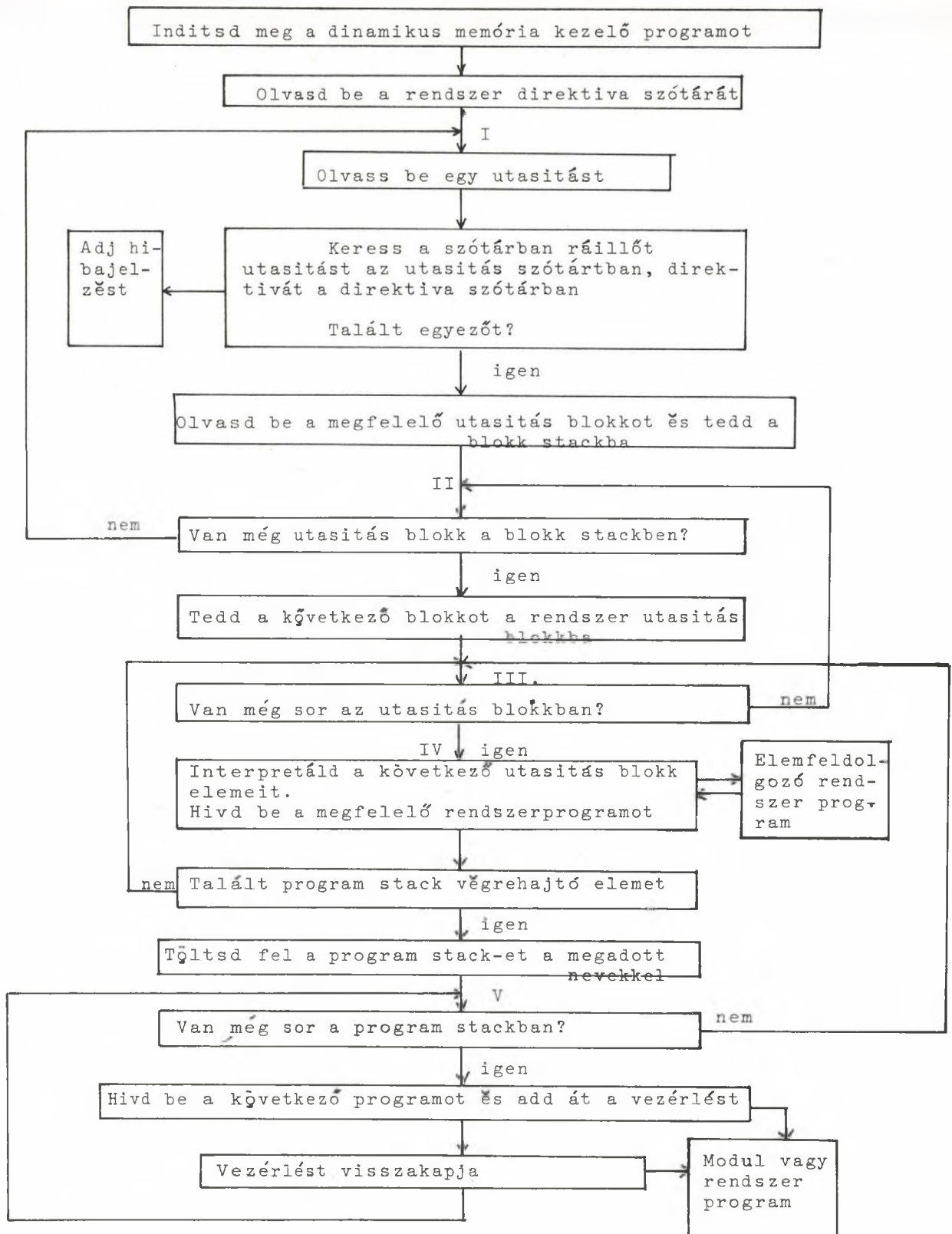
A 7. ábra vezérlési módszere mindjárt világosabb lesz, ha egy pillantást vetünk a 8. és a 9. ábrára.

A 8. ábra a monitor vázlatos működése főbb feladatainak folyamatát hozza közelebb. Először a rendszer inicializálása történik meg. Legfontosabb momentumai ennek a kisszámitógép működését optimalizáló dinamikus memóriakezelő program betöltése és a monitor rendszer egyéb tábláinak és rezidens részeinek betöltése, aktivizálása, amely a vezérlés alapszintjén az első feladat.

Ezután az utasítások vagy direktívák beolvasása következik. Ha direktívát talált, akkor a monitor direktiva szótárában keres egyezést, ha utasítást olvasott be, akkor az aktualizált utasítás szótárban keres. A munka első parancsa úgy is direktiva, amely az aktuális alrendszer szótárát behozza. Ha talált egyezést valamelyik szótárban, akkor a megfelelő utasítás blokkot betölti az utasításblokk stackbe. Ezeket a tevékenységeket az I vezérlési szint irányítja. A vezérlés szintvezérlője mindig ennek a szintnek adja a vezérlést, ha a szintvezérlés táblájában nincs más szint megjelölve.

A monitor rendszer állandóan vizsgálja, hogy a feladatok végrehajtását koordináló utasítás blokkokból van-e még és hogy ezek következményei adnak-e újabb feladatokat. Ezt a tevékenységet azzal kezdi, hogy vizsgálja van-e fel nem dolgozott utasítás blokk az utasításblokk stackban. Ha van, betölti a rendszer üres utasítás blokk helyére. Ha nincs újabb utasítást olvastat be, azaz visszatér az I szintre. Az itt leírt tevékenységet a II vezérlési szint irányítja.

Ezután a monitor vizsgálja, hogy van-e még olyan utasítás elem, amelyet nem dolgozott fel. Ez a III. vezérlési szinten történik. Ha van még elem, továbbmegy a IV. szintre, ha nincs,



8. ábra

A monitor vezérlés működése

akkor visszatér a II. szintre, hogy megnézzé van-e még utasítás blokk ezek stackjében. Ha az is kifogyott, akkor olvas be újabb utasítást kívülről.

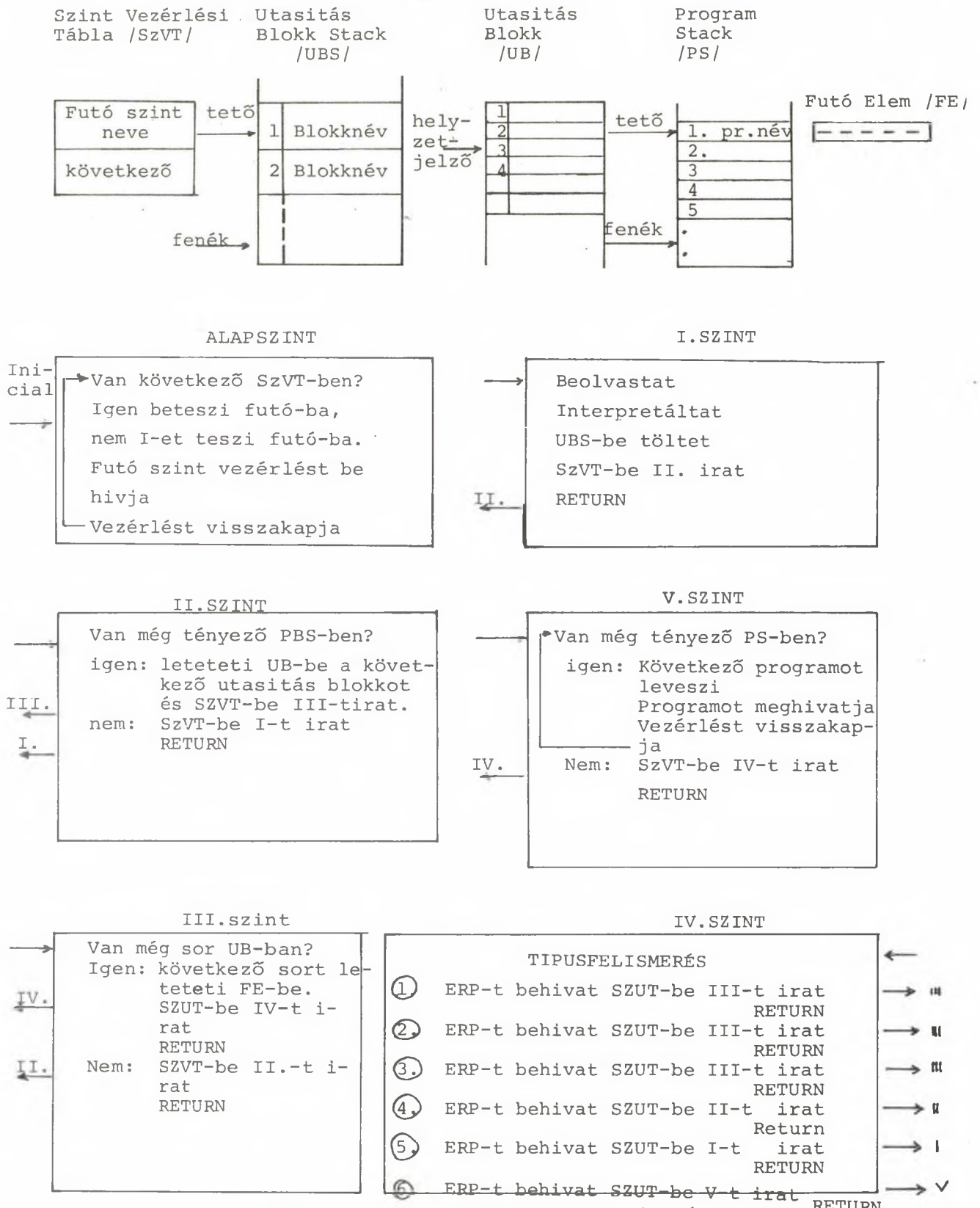
A IV. szinten folyik a rendszer üres utasítás elem helyére letett utasítás elem feldolgozása. Az elem értelmezése után a vezérlést a monitor, vagyis annak megfelelő része átadja az illető elemet feldolgozó rendszerprogramnak. Az feladát elvégezve a vezérlést visszaadja. Ha nem volt az elem modul, vagy program végrehajtást előíró típusu, azaz nem töltötte a program stacket, akkor a vezérlés visszakerül a III. szintre. Ha olyan volt, akkor az elemben megadott nevekkel az előirt sorrendben feltölteti a monitor a program stacket és átadja a vezérlést az V. szintnek.

Ezen az utolsó szinten folyik a program stack állandó vizsgálata, nem ürült-e ki már. Amíg a stackban van tényező, a monitor végrehajtó egysége átadja a vezérlést a soronkövetkező programnak, majd újra vizsgálja van-e még végrehajtandó program. Ha ez a stack kiürült, a vezérlés a III. szintre adódik át.

A 9. ábrán pontosabb ábrázolását láthatjuk a monitor vezérlési strukturájának. Mindenek előtt fordítsunk néhány szót az alkalmazott monitor rendszer elemekre.

A szintvezérlési tábla tulajdonképpen néhány szó, vagy rekord. Az elsőben az éppen futó szint neve, azaz száma van, a másodikban a következőé. Maga a szintvezérlő ezt a táblát figyeli, ha a következőben van tényező, akkor a vezérlés azon szintjét indítja. Ha nincs megnevezett mindig az I. szintet hívja.

Az utasítás blokk stack fogalma nem található meg a célrendszerekben. Örömmel hallottam viszont Tallinban, hogy ők is foglalkoznak hasonló gondolattal. Ugy tűnik, ennek beve-



9. ábra
Monitor vezérlési szintjei

ERP= Elemfeldolgozó
Rendszer
Program

zetése nagy mértékben elősegitheti a monitor rendszer rugalmasságának növelését és hatásos eszközt nyújthat a rendszerprogramozók kezébe különleges vezérlési struktúrák megoldásához.

Ez a stack hasonló felépítésű, mint a program stack. Egyszerű FILO /first in last out/ lista. A monitor rendszer megfelelő stack kezelő programja mindig a lista tetejére teszi az éppen beadott tényezőt betöltéskor. Ezt úgy éri el, hogy egy tetőjelző mutatót /pointert/ mozgat a listán. Ez mindig az éppen betöltött tényezőre mutat. Hogy e tényező egy egész utasítás blokk, vagy egy program neve, az teljesen közömbös. Kivétekelkor mindig azt a tényezőt veszi ki, amelyre a tetőjelző mutató mutat és utána a mutatót automatikusan visszalöki egy hellyel. Azaz törli a lista tetejéről a már kivett nevet.

Van egy fenék mutatója is e listának. Amikor a tető mutató megegyezik a fenék értékével a lista, vagyis a stack kiürült. Ez az egész technika semmi újat nem tartalmaz, szokásos megoldásu itt is. Utasítás blokk stack esetén nem célszerű a teljes blokkokat tárolni, inkább csak a blokkok hozzáférését kell biztosítani. Ezért a blokk nevét, vagy megfelelő mutatóját kell a stackban elhelyezni.

Az utasítás blokk itt egy üres tábla a monitor rendszerben, amelyet aktualizálni kell. Az éppen behozandó utasítás vagy direktiva blokkja kerül erre a helyre.

A futó elem szintén egy üres rekord a rendszerben, amelybe be kell tölteni az aktuális utasítás elemet.

Külön kell szólni az utasítás elemek típusairól. A különböző típusu utasítás elemekkel lehet a szintvezérlési sorrendet befolyásolni, ezen keresztül a monitor és így a rendszer munkáját a kívánt módon irányítani.

Az 1. elemtípus makrónév jellegű, egy utasítás elem sorozatot képvisel. Hatására az utasítás blokkba másolódik e makrónak megfelelő utasítás elem sorozat a megfelelő helyre. A vezérlés a III. szintnek adódik át.

A 2. elemtípus nem okoz itt változást, a vezérlés simán visszatér a III. szintre.

A 3. elemtípus az utasítás blokk sorrendjét befolyásolja. Itt jegyezzük meg, hogy az utasítás elemekkel hasonló eljárásokat teszünk lehetővé, mint a szokásos programozásban. Ugyanúgy lehető lesz az elemekre nézve ugró utasításokat, feltételes elágazokat, esetleg ciklusokat szervezni, mint programozás esetén. Ezeket képviseli a 3. elemtípus azzal, hogy a mögötte álló program az utasítás blokk mutatóját mozgatja és így megváltoztatható a sorrendiség. A vezérlés ezután is a III. szintre adódik át.

A 4. elemtípus az utasítás blokk stacket befolyásolhatja új blokk vagy blokkok berakásával, esetleg törlésével. A vezérlést a II. szintre adja át.

Az 5. elemtípus a szint vezérlő táblát befolyásolhatja. Például feladata végrehajtása során meghatározott perifériáról beolvastat új utasítást. Ekkor a vezérlést átadja az I. szintre.

A 6. elemtípus az, amelynek feladata a program stack töltése és végrehajtásának kezdeményezése. Ezzel nevezzük meg a végrehajtandó modulokat, vagy a végrehajtandó rendszerprogramokat. Általános érvénye minden program meghívására kiterjed, amelyhez a rendszer hozzá tud férni. Ezen elemtípussal bármilyen program futtatása kezdeményezhető, akár egy egész NC processzoré is, feltéve, hogy az a rendszer programozó nyelven íródott, ennél fogva a monitor dinamikus memóriakezelő része fogadni tudja. A 6. elemtípus után a vezérlés az V. szint-

nek adódik át.

Megjegyezzük még, hogy lesznek a rendszerben olyan utasítás elemek, amelyek erről a helyről is lehetővé tesznek kommunikációt a kezelővel, a felhasználóval. Ez azonban a szint vezérlést úgy tűnik külön nem befolyásolja.

Másik külön megjegyzésünk, hogy a rendszer biztosítja a modulok számára, így rendszerprogramok számára is, hogy programból hozzáférjenek a program stackhoz. Ezért a programozó nyelvbe a program stackhoz való hozzáadás, törlés, kihagyás, összekötés, másolás megoldására utasításokat kell biztosítani. A nagyszámítógépes rendszerek ezt a lehetőséget igen bonyolult módon az operációs rendszerek kijátszásával tudják csak megoldani a már említett megfejelt FORTRAN segítségével.

3.2 A monitor rendszer fő részei

Nézzük meg a monitor rendszer főbb részeit, és azok elhelyezését mutató 10. ábrát.

A monitor rendszer tulajdonképpen futtatási rendszer és ezért a szokásos operációs rendszer feladatok jó része nem szükséges akkor, amikor az egységes tervező rendszert a monitor rendszer alatt használjuk. Jobbára az operációs rendszer periféria kezelő részeire van leginkább szükség. A monitort úgy kell kialakítani, hogy ehhez hozzá tehető legyen. Ennek feltétele a meglévő operációs rendszer átszabása, vagy megfelelő méretű és részfeladatokat ellátó új operációs rendszer írása. Egyik feladat sem lebecsülendő, de kisszámítógépek esetén kézenfekvő és megoldható. Nagyszámítógépen erre gondolni sem lehet. Ott meg kell alkudni a meglévő operációs rendszer korlátokkal, esetleg azok körülményes, részleges kijátszásával, a feladat nagysága miatt.

Másik operatív memória rezidens része a monitornak a dinamikus memória kezelő program, amely kiszzámítógépek esetében, mint [9] igazolta megoldja a memória hatékony kihasználását programokra és adatokra egyaránt. Nagyszámítógépen ilyen megoldást esetleg csak adatokra lehet alkalmazni.

Állandóan az operatív memóriában kell lenni a monitor vezérlési mechanizmusának, az azt kisegítő tábláknak, stackeknek. A nagy monitor táblák, mint például normálalak tábla, változó leírás tábla, a nagy monitor fájlok mint például SQL, SQ2 buffereit szintén az operatív memóriába kell helyezni. Külön kérdés, hogy a direktiva szótár, vagy esetleg az utasítás szótár - ha elég kicsi - nem lehet-e szintén memória rezidens, ha a működtetés körülményei ezt megkívánják.

A monitor maga is moduláris felépítésű, feladatai megoldására jól tagolt modulokkal rendelkezik. Így egy-egy részfeladat ellátására mindig az aktuális modul hívódik be.

Ezenkívül a monitor rendelkezik egy sereg olyan monitor rutinnal, amelyekhez minden program modul hozzáférhet. Az alrendszerek modulirói szabadon használhatják ezeket, csakugy, mint a rendszer egyéb programjai.

A rendszerprogramok készlete elsősorban az utasítás elemfeldolgozó programokból áll, de ezt a készletet egyéb rendszerprogramok is bővíthetik.

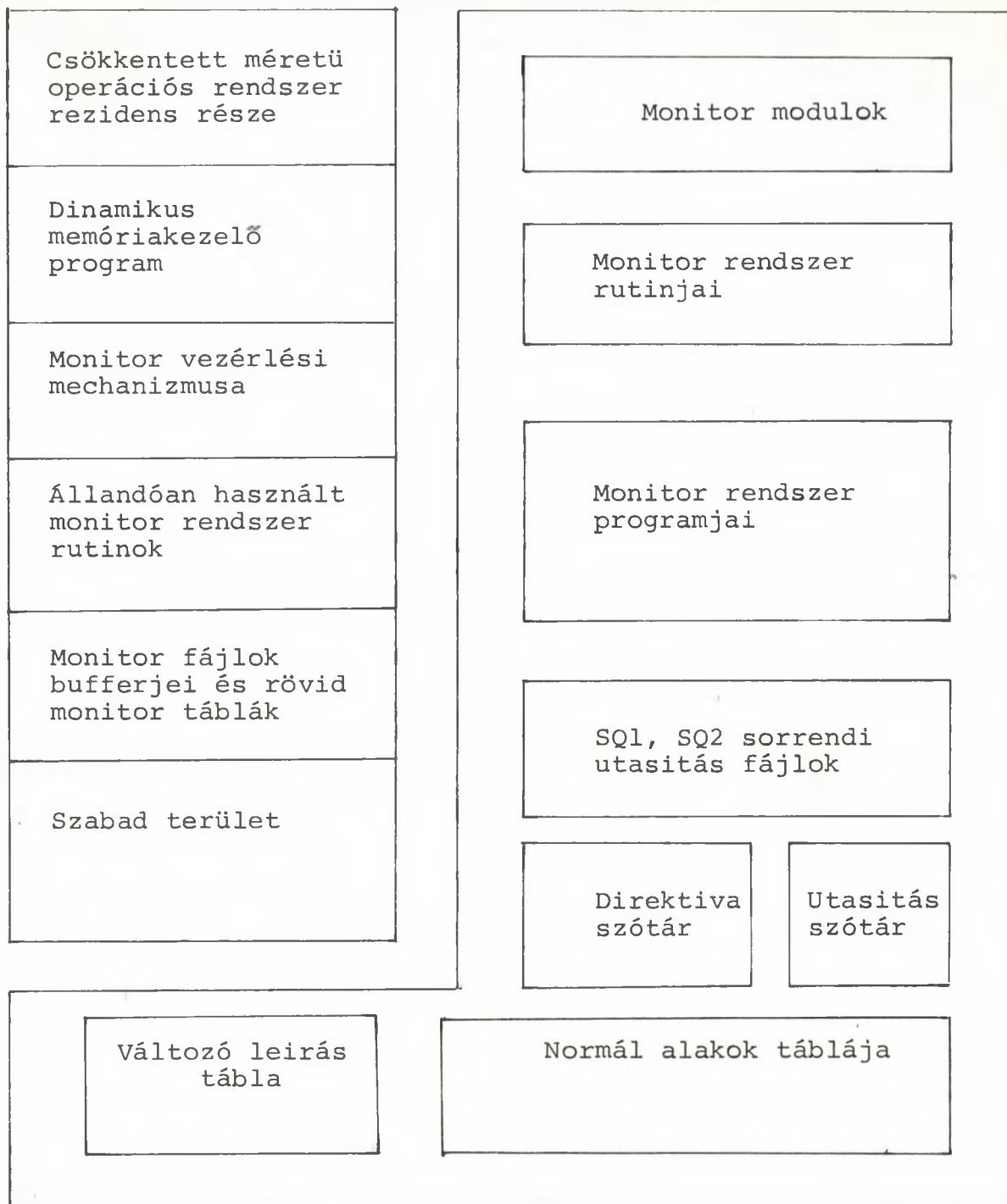
3.3 A monitor rendszer használata

Nézzük meg az alkalmazó oldaláról, milyen igények kielégítésére van lehetőség.

A rendszer irányítására szolgáló perifériák jellegzetes tulajdonságai szerint két alapvető működtetést lehet elképzelni.

OPERATIV MEMÓRIA

HÁTTÉR TÁROLÓ



10. ábra

A monitor rendszer elhelyezkedésének
vázlata a kisszámitógépen.

Az egyik a konzolirógépnek megfelelő, de nem okvetlenül konzolirógépen végzett munka. Ekkor a monitor direktivákat és a kisegítő alrendszer utasításait, de a többi alrendszer utasításait is úgy kell megadni, hogy minden karaktert be kell gépelni a billentyűzeten.

Alfanumerikus display esetében - grafikus display esetében is - megvalósítható a menüből való választás lehetősége. Ez sokkal gyorsabb. Egy-két mozdulattal választható az alap utasítás és paramétereire is rákérdezhet a számítógép. Így megoldható a tanító jellegű gyors és ami még fontosabb kevesebb hibalehetőséget magában rejtő irányítás. Ilyen jellegű párbeszédet az utasítás elemekkel be lehet programozni.

Kivánatos lehet e két működési mód keverése is, például a kisegítő alrendszer esetében, ha egyébként az első mellett törnénk lándzsát általában. Itt szövegszerkesztés esetén ég és föld a különbség kényelem és sebesség tekintetében a képernyős módszer javára.

Vizsgáljuk meg a lehetőségeket a feladatok oldaláról. Három fontos esetet nézünk meg röviden, futtatások előkészítését, alrendszer futtatását, program futtatást.

A rendszer kisegítő alrendszere révén alkalmas futtatási feladatok előkészítésére. Ezek közé tartoznak:

- fájl kezelési műveletek, mint másolás, rendezés, könyvtározás, fájlok lekötése, törlése, hozzárendelése, stb.
- szerkesztési műveletek, mint szövegek javítása, törlése, bővítés, beszúrás, stb.
- egyéb műveletek, mint leolvasás, kiiratás, lapozás, rollozás stb.

A kisegítő alrendszerben minden ilyen tevékenységre szolgál egy megfelelő utasítás. Képernyős módszer esetében csak a cursorral kell választani ezeket menüből vagy kihasználhatók a display billentyűzet speciális nyomógombjai.

Az alrendszerek futtatását bőségesen is tárgyaltuk. Láttuk, hogy végül is akár inkrementálisan működik a rendszer - tehát utasításoként olvas be, interpretál és hajt végre -, akár kötetelve kezdi a beolvasást és interpretálást, a végrehajtás már mondatonként megy a teljes feldolgozásig. Nagy mélységű többszintes programok esetén, mint amilyen a moduláris NC programozási rendszer, ez nem jó megoldás. Azonban ilyen esetekre is található módszer, még alrendszer szerű szervezésben is. Ilyenkor a többszintes program egyes szintjeit tekintjük alrendszernek, de ezen alrendszerek között megőrizzük a normál alak tábláját és a változó leírás táblát, valamint előre lekötjük a többi csatlakozási felület fájljait is az egymás közti átadásokhoz. Ez azonban nem elég. Biztosítani kell, hogy az első szint után már ne működjön a beolvasó és interpretáló egység, hanem SQ2-ről kezdődjön a munka. Azt is biztosítani lehet, hogy az alsóbb régiók alrendszerei ha kell hozzányulhassanak az SQ fájlhoz, mintegy előszerkesztve azt a következő alrendszernek. Az sem jelent különösebb nehézséget, hogy SQ olvasása esetén, ha olyan utasítást talál az éppen bent levő alrendszer, amely nem neki szól, egyszerűen átugorja, ahelyett hogy hibát jelezzon. Egyszóval a moduláris NC programozó rendszer kisszámitógépes megoldásához is jó keretet tud nyújtani az egységes tervező rendszer.

Ez a rendszer egyszerű programfuttatásokra is kiválóan alkalmas. A kisegítő alrendszer programfuttató feladatcsoportja egy utasításának megadjuk a program nevét és a futtatás egyéb paramétereit. A futtatásra szolgáló utasítás blokk elemei alapján kívánságra még egy sereg intézkedés történhet. Ezekben megadhatók eredetileg is adatok input perifériára, output perifériára, stb., de kérhet innen operátor közbelépést, például

az előző adatokat lekérdezheti az operátortól. Tetszőleges futtatás utáni kijelzés, egyéb adminisztráció, vagy más tevékenység is elvégezethető, hiszen az utasítás blokkban a futtatandó program nevén kívül sok más tevékenység megnevezésére akár program sorozatok futtatására is van lehetőség.

Igy például ez a módszer kiválóan alkalmas a moduláris rendszerbeli többszintes komplett NC processzorok futtatására. Ekkor az erre a célra szolgáló utasítás blokk úgy fogható fel, mint a moduláris rendszer eredeti rendszerterve szerinti szintvezérlő. Csak annál jóval rugalmasabb. Ebben az utasítás blokkban az első szint behívása előtt még elhelyezhetők mindenféle előkészítő műveletekre szolgáló elemek, kezelő megkérdezések inputra, outputra, fájl lekötések, stb. Ezután behívja az első szint programját az input-dekódot egy elem.

Ezen elem mögé beszurható olyan, amely a kezelő akaratától függően folytatást vagy megszakítást tesz lehetővé, hibalista kiadást kezdeményez, vagy más tevékenységet, ha szükséges.

Ezután folytatólagosan a következő elem megint programbehívó, amelyik a következő szint programját a geometriát indítja el. Ennek befejeztével a rendszer a következő elemre megy, amely megint lehet egy felhasználói véleményadást kezdeményező, és így mehet ez tovább minden szintre nézve.

Az utasítás blokkokkal arra is lehetőség van, hogy a rendszer megkérdezze melyik szinttől kezdje a program a futtatást. A megjelölt szint igényeit ugyanugy közli a kezelővel, mintha az önálló program lenne. Azt azonban figyelembe kell venni, hogy itt sem minden automatikus magától. Az egész módszert előre meg kell tervezni az indító utasításhoz tartozó utasítás blokk megtervezésekor. Igaz viszont, hogy ez utóbbi tevékenységet az alrendszer generáló nyelven leírhatjuk.

4. KÖVETKEZTETÉSEK

Az előzőekből kitűnik, hogy egységes tervező rendszer kiszámítógépekre is létrehozható. A teljes rendszerre vonatkozó részletes rendszer tervnek tartalmaznia kell:

- az alrendszerek részletes rögzítését,
- a bemenő nyelv részletes meghatározását,
- a csatlakozási felületek pontos definiálását,
- a programozó nyelv rendszertervét,
- az adatbázis és adatbázis kezelő alrendszer, valamint adatbázis kezelő nyelv rendszertervét,
- a monitor rendszer, alrendszer generáló alrendszer és kisegítő alrendszer rendszertervét.

A monitor rendszer részletes rendszertervében meg kell határozni, el kell készíteni:

- a monitor direktívák készletét,
- e direktívák blokkjait,
- az utasításelemek készletét,
- az utasítás elemeket feldolgozó programok vázlatát,
- az egyéb rendszerprogramok vázlatát,
- a dinamikus memória kezelő rendszer vázlatát,
- a monitor vezérlés vázlatát,
- a monitor modulok vázlatát,
- a monitor táblák részletes meghatározását,
- a monitor részletes folyamat ábráit.

IRODALOMJEGYZÉK

1. János J. - Dr. Horváth M. - Kovács M. - Molnár E. - Zitás I.:
A gépipari AMT tudományos főirányai és alapelvei
AMT tanulmány, MTA-SzÁTI, Budapest, 1975.május
2. Dr. Kovács I. - Andics E. - Bolla L. - Czenthe Sz. - Frank J. - Hajós Gy. - Dr. Hatvany J. - Hazai J. - Kovács M. - Lajtai I.:
Teendők diszkrét gyártó folyamatok integrált anyag és adatfeldolgozó rendszereinek bevezetésére az V. ötéves tervben
OMFB tanulmány, 4-7401-T, Budapest, 1975.julius
3. Czenthe Sz. - Cser I. - Gyürki J. - Dr. Hajdu Gy. - Dr. Horváth M. - Kasza R. - Kovács M. - Krammer G. - Dr. Somló J.:
A termelés műszaki előkészítésének automatizálása a gépiparban
AMT tanulmány, MTA-SzÁTI, Budapest, 1974.december
4. D. Ross: Ices System Design
MIT Press, Cambridge, Massachusetts, 1966.
5. B.G. Tamm: Integrirrovannaja Szisztéma Programirovvanija
INEUM intézeti közlemények 43. sz. Moszkva 1974.
6. Berta M. - Bródy F. - Cser I. - Gyürki J. - Dr. Horváth M. - János J. - Juhász M. - Kovács M. - Krammer G. - Olajos I.
A moduláris felépítésű egységes gépi programozási rendszer egységesített elemeire vonatkozó előzetes specifikáció
MNK javaslat, SPE, Budapest, 1975. január

7. Gerhardt G.: Transzportábilis kisszámítógépes programozási nyelv rendszerterve
MTA-SzTAKI tanulmány, Budapest, 1975. dec.
8. János J.: Kisszámítógépes input-dekód rendszer terve
GTI tanulmány, Budapest, 1975. december
9. Bródy F.: Virtuális memóriakezelő program
MTA-SzTAKI belső anyag
10. Cser I.: Forgácsolótechnológiai tervezőrendszer
egységes belső közbelső nyelve
GTI tanulmány, Budapest, 1974.
11. Gyürki J.: Adatbázis kezelés kisszámítógépen
MTA-SzTAKI 1975. /publikálás alatt/

"A" FÜGGELÉK

R10-es számítógép eszköz- és program- kérdéseivel kapcsolatos megfontolások

Az R10 számítógép alkalmas arra, hogy az egységes kisszámitógépes technológiai tervező rendszert rátelepítsük. Ez azonban csak akkor képzelhető el, ha megfelelő számítógép konfigurációt alakítunk ki. Egy a célnak megfelelő számítógép konfigurációt mutat be a csatolt 11. ábra.

A cél meghatározásakor nemcsak a tervező rendszer igényeit tekintettük. Egyik alapvető feladat a gép önálló használata a tervezőrendszer felépítése előtt. Ekkor megszokott módon a megszokott nyelven kell használni, tehát elsősorban FORTRAN programok futására és fejlesztésére kell alkalmasnak lennie. Ugyanakkor alapvetőnek látszik, hogy képes legyen nagyszámítógépes munkák előkészítésére.

Ezek nagyszámítógépre vitelére és a nagyszámítógépes eredmények esetleges továbbfeldolgozására is alkalmasnak kell lennie. Például gondolunk itt NC processzorok bemenő nyelvű programjainak a kisgépen való, szintaktikailag helyes megírására, nagy gépen való NC processzor futtatásra és a CLDATA kisszámitógépen való kirajzolósos ellenőrzésére, vagy posztprocesszállására.

Cél a megbízható működés is. Kisszámitógépek esetében ez leginkább a perifériákon mulik, mivel a központi egységek és operatív memóriák sokkal megbízhatóbbak. Ezért érdemes törekedni jóminőségű perifériák beszerzésére, amelyeket a legtöbb esetben csak konvertibilis valutáért lehet beszerezni, de nem mindegy, hogy a baráti gyártmányok közül melyiket választják.

A megbízható működés feltétele még a megfelelő kezdeti gép-felállítás is. Ajánlatosnak tartjuk ezt tapasztalt szakértőre

bizni, aki képes hibátlan konfiguráció átvételére. Ezért a végleges konfiguráció megtervezésétől a költségvetés összeállításán át a géprendszer átvételéig tartó munkára olyan tanácsadó szükséges, aki már több R10-es konfigurációt végig vezetett az átvételig és üzemeltetett is.

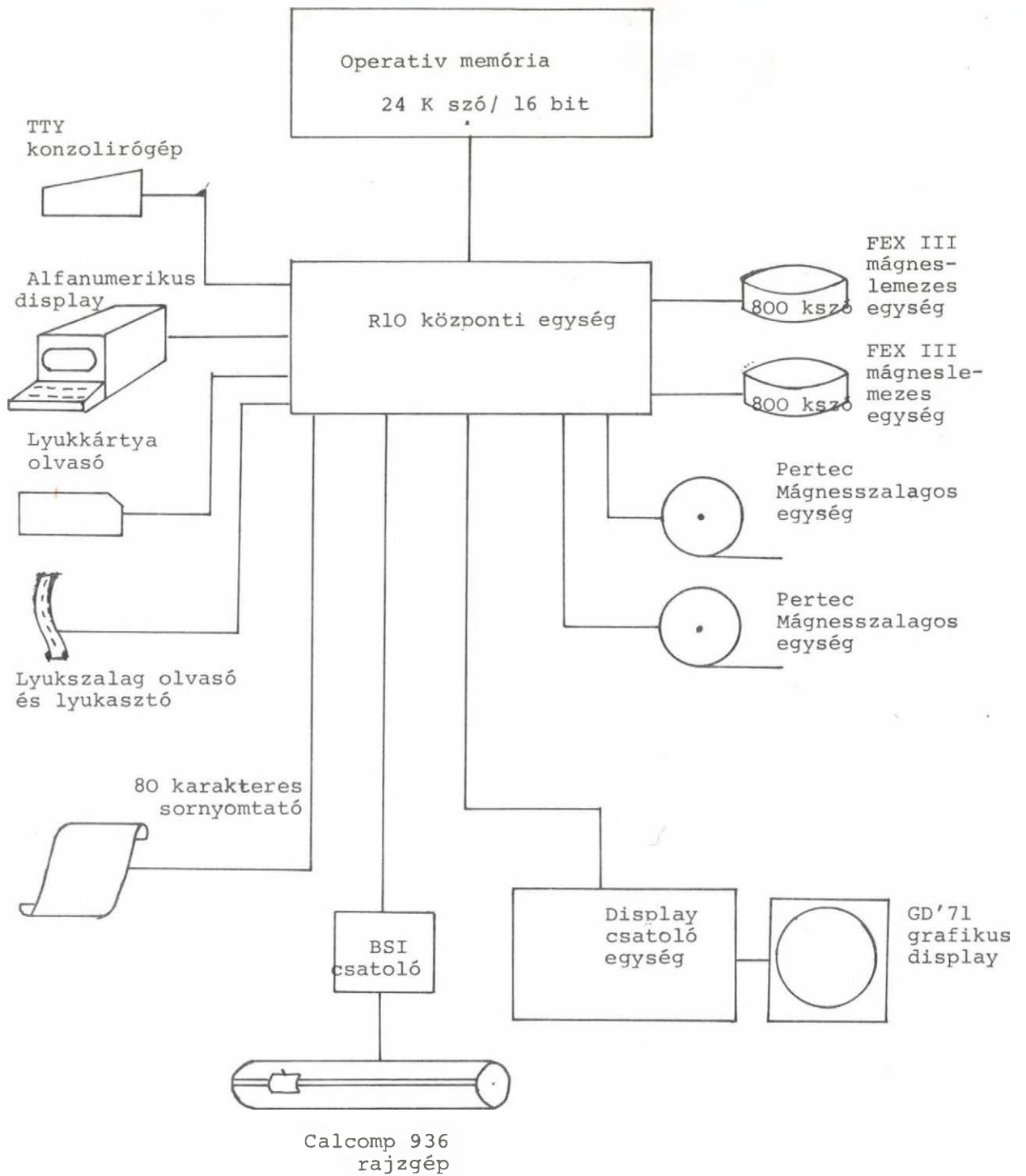
Fontos cél még az adatelőkészítés munkájának minél olcsóbb, egyszerűbb és megbízhatóbb elvégzése. A számítógép összeállítás ezt is alapvetően befolyásolja.

Vegyük sorra a 11. ábra egységeit. Ezek közül nem mindegyik elsőrendű fontosságu és nem is valósítható meg esetleg az első menetben.

A központi R10 számítógép operatív memóriájára a 24K szó/16-bit látszik a legalkalmasabbnak. Ha meggondoljuk, hogy a szállított operációs rendszerek az 5-8 K szó memóriarezidens helyfoglalásuak, a MITRASZ assembler kb 7K szó, a FORTRAN fordító program 10K szó, világos, hogy a legminimálisabb memória, amellyel FORTRAN program futtatható 16K szó terjedelmű. Később pedig, amikor tervező rendszerünk monitorjának rezidens részei is helyet foglalnak a memóriában és technológiai adatbázist is kell kezelni, esetleg a 24K szó már szűk is lesz. Most a szakértők úgy becsülik, hogy ez a 24K szó már hosszabb távra is megfelelő.

Elsőrendű fontosságu mágneslemezes gyorstárak alkalmazása. Enélkül nagy programok futtatása el sem képzelhető, de még egyszerű FORTRAN programé sem, hiszen a fordítóprogram csak a diszk közbejöttével tud kisgépen megélni, oda rakja le a programot és onnan veszi könyvtárának rutinjait is, más fájlokról nem is beszélve.

A megfelelő diszk kiválasztása már nehezebb feladat. Látványlag előnyös lenne cserélhető lemezes egységet használni. Ennek azonban két dolog mond ellent. Az egyik, hogy a cserél-



11. ábra

R10 tervezői konfiguráció

hető lemezes egységek mozgófejesek és valóban megbízhatatlanab-
bak ezért az állófejeseknél. A másik, hogy az állófejes egy-
ségek 5-10-szer gyorsabb hozzáférések mint a mozgófejesek. I-
gaz viszont, hogy a hozzáférhető cserélhető lemezen az egysé-
gek háromszoros tároló kapacitásuk, mint a fix fejesek. Mind-
ezek alapján két fix fejes gyors hozzáférésű diszket javaso-
lunk. Az R10 ezeket úgy tekinti, mint egymás folytatásai len-
nének és csak egy nagyobb diszk egysége lenne. Mivel a fix fe-
jes olcsóbb is, ez a megoldás kézenfekvő. A duplázás egyben
tartalékot is jelent. Ugyanakkor gyors diszk mellett nem prob-
léma nagy időközökben mágnesszalagról tölteni.

Javasolunk legalább két mágnesszalagos egységet. Kettőt,
hogy adatrendezési, fájlkezelési, adatbázisbeli munkákra is
alkalmas legyen. Ezenkívül melegtartalék szerepe is van, de
ez másodlagos. Felhívjuk a figyelmet arra, hogy a szállító cég
szeret igen rossz minőségű mágnesszalagos egységeket ajánlani.
Ezek saját számítóközpontjukban sem működnek megbízhatóan. E
témában is feltétlenül fontos a szakértő tanácsadása.

A mágnesszalagos egységnek olyannak kell lennie, hogy nagy-
számítógép kompatibilis mágnesszalaggal dolgozzon. Erre külö-
nösen azért van szükség, hogy előkészített munkát nagygépre
lehessen vinni egyszerűen és elkészült eredményt, részered-
ményt a kisgépen lehessen hasznosítani.

Bemeneti egységként ajánlatos a konzolirógép, és ma már
nem hagyható el az alfanumerikus display sem. Lyukkártya olva-
só szükségesnek látszik, annál is inkább minthogy a gép ere-
deti programjai kártyarendszerre készültek. Lyukszalagolvasó
és lyukasztó ugyan egyike a legmegbízhatatlanabb perifériák-
nak, de áruk és a használt anyag összemérése alapján mégis
nélkülözhetetlennek tűnik. Ehhez az indokhoz jön még az NC
szerszámgépek esetleges lyukszalag igénye. Sornyomtató minden-
képpen kell, de megfelel a 80 karakter hosszúsága is, amely

éppen egy lyukkártyányi sort képes kiírni.

Az egyik grafikus periféria lehet rajzgép. A Calcomp 936-os típus elég gyors, elég pontos és elég olcsó is, igaz devizás. Vezérlés nélkül javasoljuk beszerezni és egy BSI csatolóval a géphez kötni. Ilyen megoldást a SzTAKI-ban már készítették.

A másik grafikus, egyben interaktív periféria lehet egy GD'71 típusú grafikus display. Ezt csatoló egység köti az R10 számítógéphez. A csatoláshoz szükséges különböző berendezéseket összefoglalva egy blokkal szemléltettük a 11. ábrán.

A GD'71 display - R10 számítógép csatolásának eszköz és program oldalával kapcsolatos munkák folyamatban vannak. Azt azonban figyelembe kell venni jóelőre, hogy a grafikus display használatakor magának a grafikus displaynek a kiszolgálása körülbelül 8K szó központi memória területet igényel.

A grafikus display csatlakoztatása a tervezői elrendezéshez természetesen másképpen is elképzelhető. Ha a 11. ábrán a csatoló egység helyére egy másik, de csupasz kissetszámítógépet képzelünk 8K szó memóriával, akkor egy olyan összeállítást kapunk, amelyhez hasonló a TKI-ban kísérleti üzemben működik. Ennek előnye, hogy a tervezői konfiguráció rugalmasabb használatát teszi lehetővé. Azonban az ilyen megoldások eszköz és program oldalai még nincsenek kicsiszolva, teljesen megoldva. A TKI-ban is egy R10-es géphez TPA géppel van a grafikus display csatolva. Karbantartás és mindenkor tartalék szempontjából előnyösebbnek látszik a vázolt megoldást két R10-es számítógépre alapozni. Mindenesetre a közeljövőre nézve mi az előző bekezdésben leírt és az ábrán látható megoldást javasoljuk.

A programokról szólva említettük, hogy az R10 kissetszámítógépre van operációs rendszer, assembler és FORTRAN fordítóprogram. Ezen kívül a gyártó cég még szállít BASIC programot

is. A választékot még kiegészíti néhány célorientált, nem gépészeti jellegű feladat megoldására alkalmas programrendszer.

A tanulmányban vázolt tervezőrendszer igényeit a géppel együtt vehető programok nem elégítik ki, de mindenesetre arra jók, hogy a gépet azonnal a felállítást követően használhassák.

A tervezőrendszer céljára sokkal alkalmasabbnak látszik az intézetünkben kifejlesztett IDOS nevű, alfanumerikus display-re alapozott, interaktív feladatokra orientált operációs rendszer alkalmazása. Ennek állandó helyfoglalása az R10 operációs memóriájában mindössze 2K szó. Nagyon valószínű, hogy az IDOS-t a tervező rendszer igényeinek megfelelően tovább kell fejleszteni. Az is lehetséges, ha az a járhatóbb út, hogy az eddigi tapasztalatokon és szakértelmen alapuló új operációs rendszert kell írni, amely jól illeszkedik a megírandó monitorrendszerhez. A tervező rendszerhez azonban mindezek előtt, vagy mellett a [7] -ben vázolt programozónyelv R10-es implementálására van szükség.

A TANULMÁNYOK sorozatban eddig megjeletek:

- 1/1973 Pásztor Katalin: Módszerek Boole-függvények minimális vagy nem redundás, $\{\wedge, \vee, \neg\}$ vagy $\{\text{NOR}\}$ vagy $\{\text{NAND}\}$ bázisbeli, zárójeles vagy zárójel nélküli formuláinak előállítására
- 2/1973 Башкеви Иштван: Расчленение многосвязных промышленных процессов с помощью вычислительных машин
- 3/1973 Ádám György: A számítógépipar helyzete 1972 második felében
- 4/1973 Bányász Csilla: Identification in the Presence of Drift
- 5/1973* Gyürki J.-Laufer J.-Girnt M.-Somló J.: Optimalizáló adaptív szerszámgepirányítási rendszerek
- 6/1973 Szelke E.-Tóth K.: Felhasználói Kézikönyv /USER MANUAL/ a folytonos Rendszerek Szimulációjára készült ANDISIM programnyelvhez
- 7/1973 Legendi Tamás: A CHANGE nyelv/multiprocesszor
- 8/1973 Klafszky Emil: Geometriai programozás és néhány alkalmazása
- 9/1973 R.Narasimhan: Picture Processing Using Pax
- 10/1973 Dibuz Á.-Gáspár J.-Várszegi S.: MANU-WRAP hátlaphuzalozó MSI-TESTER integrált áramköröket mérő, TESTOMAT-C logikai hálózatokat vizsgáló berendezések ismertetése
- 11/1973 Matolcsi Tamás: Az optimum-számítás egy új módszeréről
- 12/1973 Makroprocesszorok, programozási nyelvek. Cikkgyűjtemény az NJSzT és SzTAKI közös kiadásában. Szerkesztette: Legendi Tamás
- 13/1973 Jedlovsky Pál: Új módszer bonyolult rektifikáló oszlopok vegyész-mérnöki számítására
- 14/1973 Bakó András: MTA kutatóintézeteinek bérszámfejtése számítógéppel
- 15/1973 Ádám György: Kelet-nyugati kapcsolatok a számítógépiparban
- 16/1973 Fidrich I.-Uzsoky M.: LIDI-72 listakezelő rendszer a Digitális Osztályon, 1972. évi változat
- 17/1974 Gyürki József: Adaptív termelésprogramozó rendszer /APS/ termelőműhelyek irányítására

- 18/1974 Pikler Gyula: MINI-számítógépes interaktiv alkatrész-programíró rendszer NC szerszámgépek automatikus programozásához
- 19/1974 Gertler, J.-Sedlak, J.: Software for process control
- 20/1974 Vámos, T.-Vassy, Z.: Industrial Pattern Recognition Experiment - A Syntax Aided Approach
- 21/1974 A KGST I. -15-1.: "Diszkrét rendszerek automatikus vezérlése" c. témában 1973. februárban rendezett szeminárium előadásai
- 22/1974 Arató, M.-Benczur, A.-Krámli, A.-Pergel, J.: Stochastic Processes, Part I.
- 23/1974 Benkő S.-Renner G.: Erősen telített mágneses körök számítógépes tervezési módszerei
- 24/1974 Kovács György-Franta Lászlóné: Programcsomagok elektronikus berendezések hátlaphuzalozásának tervezésére
- 25/1973 Járdán R. Kálmán: Háromfázisu tirisztoros inverterek állandósult tranziens jelenségei és belső impedanciája
- 26/1974 Geregely József: Numerikus módszerek sparse mátrixokra
- 27/1974 Somló János: Analitikus optimalizálás
- 28/1974 Vámos Tibor: Tárgyfelismerési kísérlet nyelvi módszerekkel
- 29/1974 Móricz Péter: Vegyész-mérnöki számítási módszerek fázis-egyensúlyok és kémiai egyensúlyok vizsgálatára
- 30/1974 Vassy, Z.-Vámos, T.: The Budapest Robot - Pragmatic Intelligence
- 31/1975 Nagy István: Frekvenciaosztásos középfrekvenciás inverterek elmélete
- 32/1975 Singer D., Borossay Gy., Koltai T.: Gázhálózatok optimális irányítása különös tekintettel a Fővárosi Gázművek hálózataira
- 33/1975 Vámos, T.-Vassy, Z.: Limited and Pragmatic Robot Intelligence
Mérő, L.-Vassy, Z.: A Simplified and Fastened Version of the Hueckel Operator for Finding Optimal Edges in Pictures
Галло В.: Программа для распознавания геометрических образов, основанная на лингвистическом методе описания и анализа геометрических структур

- 34/1975 László Nemes: Pattern Identification Method for Industrial Robots by Extracting the Main Features of Objects
- 35/1975 Garádi - Krámlí - Ratkó - Ruda: Statisztikai és számítástechnikai módszerek alkalmazása kórházi morbiditás vizsgálatokban
- 36/1975 Renner Gábor: Elektromágneses tér számítása nagyhőmérsékletű anyagban
- 37/1975 Edgardo Felipe: Specification problems of a process control display
- 38/1975 Hajnal Andrásné: Nemlineáris egyenletrendszerek megoldási módszerei
- 39/1975^x A. Abd El-Sattar: Control of induction motor by three phase thyristor connections in the secondary circuit
- 40/1975 Gerhardt Géza: QDP Grafikus interaktív szubrutinok a CDC 3300-GD'71 grafikus konfigurációra
- 41/1975 Arató, M.-Benczur, A.-Krámlí, A.-Pergel, J.: Stochastic Processes, Part II.
- 42/1975 Arató M.: Fejezetek a matematikai statisztikából számítógépes alkalmazásokkal
- 43/1975 Matavovszky Tibor - dr. Pásztorné, Varga Katalin: Programrendszer Boole-függvény együttes egyszerűsítésére vagy minimalizálására
- 44/1975 Bacsó Nándorné: Pneumatikus áramköri hazardok
- 45/1975 Varga András: Ellenpárhuzamos félvezetőpárokkal vezérelt aszinkronmotoros hajtások számítási módszerei
- 46/1976 Galántai Aurél: Egylépéses módszerek lokális hibabecslései
- 47/1976 Abaffy József: A feltétel nélküli függvényminimalizálás kvadratikusan befejeződő módszerei
- 48/1976 Strehó Mária: Stiff típusú közönséges differenciálegyenletek megoldásáról
- 49/1976 Gerencsér László: Nemlineáris programozási feladatok megoldása szekvenciális módszerekkel
- 50/1976 Robert Treer: A syntax macro definition language
- 51/1976 Bakó András: TIMER időredukciós programcsomag

- 52/1976 W.A. Potas: Computer Aided Design
- 53/1976 Farkas Ernő: MP \emptyset .2 makroprocesszor általános ismertetése
- 54/1976 N. N. Puri: Multi Element Fault Isolation in Electronic Circuits
- 55/1976 Edgardo Felipe: The design of color, Raster-Scan graphical displays for process control applications
- 56/1976 Bán Ilona: Iterációs módszerek lineáris rendszerekre

Jelen dolgozat a 3.1-G sz.
intézeti téma keretében
került kidolgozásra

A *gal jelölt kivételével a sorozat kötetei megrendelhetők az
Intézet könyvtáránál /Budapest, XIII.Victor Hugo u. 18-22./



